

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: «Розробка 3D відеогри "Existence for a while" на базі рушія Unity»

Виконав: студент IV курсу, групи СНс-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Шубалий І. В.

(прізвище та ініціали)

Керівник

(підпис)

Дмитроца Л. П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С. В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Стоянов Ю. М.

(прізвище та ініціали)

Тернопіль  
2024

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

«    » червня 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Шубалому Ігорю Володимировичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка 3D відеогри "Existence for a while" на базі рушія Unity

Керівник роботи Дмитроца Леся Павлівна, кандидатка технічних наук, доцентка  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-472

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Літературні та інтернет джерела щодо створення комп'ютерної гри у 3D середовищі, ігровий рушій Unity.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз сфери розробки комп'ютерних ігор. 1.1 Аналіз GameDev індустрії.

1.2 Дослідження програмного забезпечення необхідного для розробки. 1.3 Основний спектр інструментів при роботі із 3D рушієм Unity. 1.4 Висновок до першого розділу.

2. Проектування відеогри у жанрі cooking simulator. 2.1 Дослідження жанру cooking симуляторів – простота і цікавість. 2.2 Огляд допоміжного інструменту Unity Asset Store.

2.3 Проектування структури комп'ютерної гри. 2.4 Висновок до другого розділу.

3. Безпосередня розробка відеогри зі всіма її компонентами. 3.1 Розробка візуальної складової відеогри. 3.2 Розробка логіки та взаємодії гравця із предметами

3.3 Розробка механізму замовлень – головна ціль гри. 3.4 Продюсування саунд-дизайн частини. Меню налаштувань. 3.5 Висновок до третього розділу. 4. Безпека життєдіяльності, Основи охорони праці. 4.1 Основні положення, стан електробезпеки в Україні

та дія електричного струму на людину. 4.2 Ергономічні вимоги до організації робочих місць користувачів комп'ютерів. 4.3 Організація і функціонування системи управління охороною праці. 4.4 Висновок до четвертого розділу. Висновок. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульний слайд. 2. Аналіз предметної області. 3. Вимоги щодо реалізації комп'ютерної гри. 4. Обґрунтування використовуваних засобів розробки. 5. Структура комп'ютерної гри

6. Ігрова мапа комп'ютерної гри. 7. Елементи інтерфейсу комп'ютерної гри. 8. Програмна реалізація логіки приймання видачі замовлень. 9. Keybinding кнопок в меню налаштувань.

10. Підготовка комп'ютерної гри до розповсюдження. 11. Подальші перспективи розвитку відеогри. 12. Заключний слайд.



## АНОТАЦІЯ

Розробка 3D відеогри "Existence for a while" на базі рушія Unity // Кваліфікаційна робота освітнього рівня «Бакалавр» // Шубалий Ігор Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2024 // С. 74, рис. – 41, табл. – 1, кресл. – , додат. – 2, бібліогр. – 37.

**Ключові слова:** game development, життєвий цикл, комп'ютерна графіка, sound design, scriptable object, анімація, ігровий рушій, рендеринг.

Кваліфікаційна робота присвячена створенню комп'ютерної гри на базі 3D-рушія Unity для навчання, розвитку навичок стратегічного мислення та розваг у реалістичних умовах.

В першому розділі кваліфікаційної роботи проаналізовано стан розвитку індустрії розробки ігор, досліджено програмне забезпечення, що використовуватиметься при розробці, розглянуто основний спектр інструментів при роботі із 3D рушієм Unity.

В другому розділі кваліфікаційної роботи досліджено жанр socking симуляторів і викрито їхню ідею в простоті та цікавості, оглянуто допоміжний інструмент Unity Asset Store задля меншої часозатратності, спроектовано структурні одиниці комп'ютерної гри.

В третьому розділі кваліфікаційної роботи описано кожен етап процесу розробки проєкту, розроблено кожен механізм із використанням технологій різних спрямувань.

**Об'єкт дослідження** виступають етапи розробки 3D-моделей об'єктів, UI-компонентів, 2D-спрайтів, музичного супроводу та ігрової механіки виготовлення замовлень для відеогри на базі рушія Unity.

**Предмет дослідження** методи розробки відеогри на базі рушія Unity.

## ANNOTATION

Development of the 3D video game "Existence for a time" based on the Unity engine // Qualification work of the educational level "Bachelor" // Shubalyi Ihor Volodymyrovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-42 // Ternopil, 2024 // P. 74, fig. - 41, tabl. - 1, chair. - , annexes. – 2, references - 37.

**Keywords:** game development, life cycle, computer graphic, sound design, scriptable object, animation, game engine, rendering.

The qualification work is devoted to the creation of a computer game based on the 3D Unity engine for training, development of strategic thinking skills and entertainment in realistic conditions.

In the first section of the qualification work, the state of development of the game development industry was analyzed, the software that will be used during development was reviewed, and the main range of tools for working with the Unity 3D engine were considered.

In the second section of the qualification work, the genre of cocking simulators was investigated and their idea was exposed in simplicity and interest, the auxiliary tool Unity Asset Store, and the structural units of the computer game were designed.

In the third section of the qualification work, each stage of the project development process is described.

**The research object** is at the is the stages of development of 3D models of objects, UI components, 2D sprites, musical accompaniment and game mechanics of making orders for a video game based on the Unity engine.

**The subject of research** are methods of video game development based on the Unity engine.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

3D (в контексті комп'ютерної графіки та даної роботи, від англ. three dimensions – три виміри) – Тривимірний простір.

DAW – (англ. Digital audio workstation) – Цифрова звуковав термінології музичних продюсерів — електронна система, призначена для запису і редагування цифрового аудіо.

HUD (англ. head-up - призначений для перегляду без нахилу голови, display - індикація) – частина графічного інтерфейсу користувача, яка дозволяє отримати різноманітну ігрову інформацію не викликаючи додаткові меню.

UI (англ. User interface) – Інтерфейс користувача для зручної взаємодії користувача (людини) з інформаційною системою.

GameDev (англ. Game development) – Розробка ігор.

Keybinding – метод оптимізації ігрового процесу шляхом налаштування елементів керування певною грою відповідно до власних потреб.

Particle system – Система частинок, яка використовується в комп'ютерній графіці способом подання 3D-об'єктів, що не мають чітких геометричних меж (різні хмари, туманності, вибухи, струмені пари, шлейфи від ракет, дим, сніг, дощ тощо).

Sprite (у комп'ютерній графіці – растрове зображення, інтегроване у більшу сцену, найчастіше у двовимірній відеогрі, але, як демонструє наш випадок, не тільки у двовимірній) – Окреме двовимірне зображення.

ВДТ – Відеодисплейний термінал.

Симулятор – Жанр відеоігор, що характеризується імітацією певної реальної речі, ситуації, чи процесу.

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1. АНАЛІЗ СФЕРИ РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР .....	10
1.1 Аналіз GameDev індустрії .....	10
1.2 Дослідження програмного забезпечення необхідного для розробки .....	15
1.3 Основний спектр інструментів при роботі із 3D рушієм Unity .....	19
1.4 Висновок до першого розділу .....	23
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВІДЕОГРИ У ЖАНРІ COOKING SIMULATOR .....	24
2.1 Дослідження cooking-симуляторів – простота і цікавість.....	24
2.2 Огляд допоміжного інструменту Unity Asset Store.....	25
2.3 Проєктування структури комп'ютерної гри.....	27
2.3.1 Проєктування початкового екрану.....	30
2.3.2 Проєктування ігрового HUD-а.....	31
2.3.3 Проєктування вікна налаштувань.....	32
2.3.4 Проєктування ігрової зони .....	33
2.4 Висновок до другого підрозділу .....	34
РОЗДІЛ 3. БЕЗПОСЕРЕДНЯ РОЗРОБКА ВІДЕОГРИ ЗІ ВСІМА ЇЇ КОМПОНЕНТАМИ.....	35
3.1 Розробка візуальної складової відеогри.....	35
3.1.1 Створення 2D-спрайтів майбутніх іконок.....	35
3.1.2 Створення базових об'єктів середовища у Blender .....	38
3.1.3 Анімація головного персонажа та об'єктів у грі .....	39
3.1.4 Розробка шейдеру для анімації Delivery Counter-а.....	42
3.1.5 Налаштування камери, світлокорекція, пост-обробка .....	43
3.2 Розробка логіки та взаємодії гравця із предметами довкілля .....	45
3.2.1 Програмування головного персонажа. Тестування рухів у ігровому середовищі.....	45
3.2.2 Розробка логіки для тумб .....	49

3.2.3	Подання інгредієнтів у вигляді скриптових об'єктів .....	49
3.2.4	Розробка UI-компонентів у грі .....	50
3.2.5	Створення сцен головного меню та завантаження. Вікно паузи .....	52
3.3	Розробка механізму замовлень – головна ціль гри .....	52
3.3.1	Тестування валідності виконання замовлення.....	54
3.3.2	Методи покращення продукту, шляхи подальшого розвитку.....	54
3.4	Продюсування саунд-дизайн частини. Меню налаштувань .....	55
3.5	Висновок до третього розділу .....	57
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....		59
4.1	Основні положення, стан електробезпеки в Україні та дія електричного струму на людину .....	59
4.2	Ергономічні вимоги до організації робочих місць користувачів комп'ютерів .....	60
4.3	Організація і функціонування системи управління охороною праці .....	62
4.4	Висновок до четвертого розділу .....	65
ВИСНОВКИ.....		66
ПЕРЕЛІК ДЖЕРЕЛ .....		68
ДОДАТКИ		



## ВСТУП

**Актуальність теми.** Нікого в сьогоднішній вже не дивує персональний комп'ютер та речі, які можна за допомогою нього створити. Але це лиш на перший погляд. Дивлячись на те, як за всього лиш тридцять років, із появою інтернету, а внаслідок цього й зі збільшенням доступу до інформації, буквально будь-кому, якщо, звичайно, маємо на увазі користувача, що зацікавлений у пошуках нового, випадає можливість стати тим, ким він чи вона хотіли б. Але, все ж не без сумлінних зусиль та бажання до змін. В цей же час індустрія розробки відеоігор розкривається все більшими масштабами для своїх майбутніх послідовників. І, як ми можемо побачити в сьогоднішній привертає все більше уваги не тільки як середовище розробки для програмістів, а вміщує в собі великий спектр спеціалістів, що стирають межі віртуальних світів аби ігри ставали частиною нашого.

Ідеї, що сплітаючись утворюють продукт – більш заземлене визначення для GameDev індустрії. Взаємозв'язок творчості, бізнесу та розчарувань. У час цвітіння нових Meta-всесвітів, тема розробки відеоігор є не тільки актуальною – з часом вона даватиме афект й на інші галузі.

**Мета і задачі дослідження.** Мета роботи полягає у розробці 3D відеогри у жанрі cooking (кулінарний) симулятор для навчання, розвитку навичок та розваг у реалістичних умовах. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати стан досліджень в індустрії розробки відеоігор.
- Дослідити програмне забезпечення, потрібне для розробки проєкту.
- Ознайомитися із основним спектром інструментів, що використовуються при розробці будь-якого жанру гри.
- Детальніше розглянути аналоги жанру відеогри, що знаходиться на стадії проєктування.

– Спроекувати прототипи об'єктів майбутнього продукту із усіма класичними для нього компонентами: інтерфейси (головне меню, HUD, меню паузи), ігрова зона, механіки взаємодії гравця із середовищем.

– Розробити кожен компонент проєкту: іконки, UI-компоненти, дизайн, рух камери, світлокорекція, 3D-моделі, анімація, шейдери, логіка взаємодії та програмування механізмів взаємодії у грі, sound design, меню налаштувань, пост-обробка, фінальний рендер.

**Практичне значення одержаних результатів.** Результати фінального продукту розробки даватимуть ряд навчальних та практичних знань у галузі розробки відеоігор, кожної їх складової зокрема, із вирішенням ряду проблем, що виникали під час виконання. Варто відзначити підвищення навичок володіння технологіями рушія Unity версії 4.6 та засобами допоміжних програм: Blender, FL Studio 20, Adobe Photoshop, Figma.

## РОЗДІЛ 1. АНАЛІЗ СФЕРИ РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР

### 1.1 Аналіз GameDev індустрії

Індустрія комп'ютерних ігор (розробка, публікація та маркетинг ігор) є однією з найбільш швидкозростаючих галузей комп'ютерних технологій і водночас сектору розваг у світі. Ігри стають феноменом культури і визнаються витворами мистецтва [1]. Кількість геймерів зростає (2,5 мільярда гравців у 2019 році), а серед вакантних місць компаній у цій галузі з'являються високооплачувані робочі місця. Виникає потужна екосистема.

Ігрова індустрія дуже особлива, включаючи інвестиційний аспект. До недавнього часу розробка комп'ютерних ігор вважалася переважно творчим процесом, що ускладнювало прогнозування комерційного успіху. Проте потік інвестицій у цей сектор зростає. З'являються венчурні фонди, що спеціалізуються на комп'ютерних іграх. За даними аналітичного агентства, капітальні інвестиції у світову ігрову індустрію у 2022 році досягли \$7,2 млрд, включаючи венчурні інвестиції та IPO (зростання на 12% порівняно з рівнем 2020 року) [2]. На рисунку 1.1 наведено діаграму річного обсягу інвестицій у індустрію відеоігор у всьому світі з 2013 по 2023 рік.

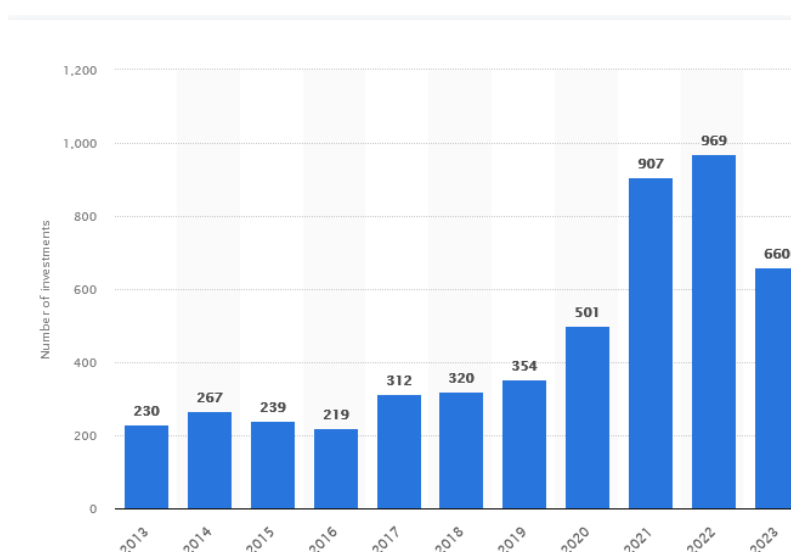


Рисунок 1.1 – Діаграма обсягу інвестицій у розробку ігор у 2013-2023 рр.

За даними Digi-Capital, найбільш цікавим напрямком для інвесторів на даний момент є технологічна інфраструктура, розрахована на велику кількість користувачів, і мобільні ігри.

Експерти вважають, що саме ігрова індустрія відповідає за сучасний креатив та інноваційні ідеї. Зараз комп'ютерний геймінг мейнстрімний і охоплює багато сфер масової культури, є її екземпляри, що стали популярними феноменами (наприклад, GTA та серія ігор Resident Evil). У США, найбільшому киту розвитку цієї галузі, комп'ютерні ігри, як не дивно, вважаються самостійним видом мистецтва. Крім того, була створена нова субкультура з власними сленгами, символами і подіями, так як і в інших. Створюється міцна екосистема. Сфера відеоігор стає роботою з хорошим доходом. За підрахунками експертів, значна частина користувачів ігор у всьому світі налаштовані зробити її основним видом діяльності.

Зараз учасники ігрового ринку поділяються на два основних типи: безпосередні розробники та видавці комп'ютерних ігор. Розглянемо ці категорії докладніше. Розробники комп'ютерних ігор – це ІТ-компанії різного розміру, які спеціалізуються на створенні ігрового контенту. Розробник може бути як фізичною особою, так і великою організацією з розподілом обов'язків між співробітниками в таких сферах, як програмування, дизайн, художні компоненти, тестування тощо [3].

Видавець — це компанія, яка займається продажем комп'ютерних ігор, створених у видавництві або окремими компаніями, які безпосередньо беруть участь у їх розробці. Тут слід зазначити, що видавничим бізнесом цікавляться не лише спеціалізовані компанії (наприклад, Valve), а й виробники електроніки (наприклад, Apple), які мають усі ресурси для виведення на ринок перспективних ігрових та інтернет-рішень. Навіть корпорації, типу Google, зацікавлені у перспективах цієї сфери.

Ігрова індустрія істотно змінилася за останні 5-7 років. Поява і популяризація магазинів мобільних додатків і цифрових платформ дистрибуції зробили процес видання ігор в світовому масштабі доступним навіть для

невеликих студій. Йдеться про так званих інді-ігри (indie games) - комп'ютерних ігри, що створюються окремими розробником або невеликою командою, як правило, без фінансової підтримки офіційного видавця комп'ютерних ігор.

Варто відзначити, що зараз великі видавці (наприклад, Sony або Microsoft) стали активно підтримувати незалежних розробників комп'ютерних ігор, надаючи їм необхідні ресурси, але не обмежуючи творчі ідеї.

Ця сфера бізнесу має чіткі перспективи, але із залученням технологій ШІ, з'явиться можливість якісніше оцінювати цифрову зрілість компаній, аналізуючи дизінформацію та неправдиві новини, захищаючи сам бізнес [4].

Вибір моделі монетизації комп'ютерних ігор залежить від платформи (консоль, смартфон/планшет або персональний комп'ютер) і регіону. В одних країнах не поширені консолі, в інших населення через високий рівень інтернет-піратства в принципі не готове платити за оригінальні цифрові продукти. Також в більшості сучасних ігор використовується гібридний підхід до монетизації - в поєднанні способи утворюють гнучку модель, де недоліки одного способу монетизації можуть забезпечувати переваги для іншого. Розглянемо всі найбільш популярні підходи [5]:

– Продажі фізичних і цифрових копій гри - класична модель монетизації (Pay2Play, P2P). За певну плату користувач отримує доступ до ігрового контенту і ігрового процесу. Як вже говорилося, зараз основні продажі ігрової продукції здійснюються на цифрових платформах. Примітно, що в ряді випадків ціни в фізичному і цифровому магазині можуть бути практично однакові - політика видавця. У цьому випадку, щоб скоротити витрати на упаковку, друк дисків, доставку в торговельні точки і т.д., споживачі набувають барвисто оформлену коробку з кодом доступу всередині замість диска. Виняток становлять колекційні видання.

– Мікротранзакційна модель. Зараз все популярнішими стають моделі низького входу (розмір платежу, як правило, не перевищує 5 дол.) або нульовою вартістю доступу з наступним продажем персоналізованих продуктів (Free2Play, F2P). Такий підхід дозволяє гравцям ознайомитися з грою і оцінити

якість продукту. Далі, після залучення в гру видавець починає заробляти на мікротранзакції. Зокрема, гравцеві пропонується заплатити за комфорт (наприклад, відключення спливаючій реклами) і можливість зміни внутрішньо ігрового статусу: преміум-акаунти з різними привілеями, придбання внутрішньої валюти, колекційного спорядження, зміна характеристик персонажа і ін.

– Продаж епізодів. На перший погляд така модель здається менш витратним способом в порівнянні з одноразовим випуском комп'ютерної гри. Але фактично використання такої моделі пов'язане з рядом складнощів. Зокрема, необхідно постійно підтримувати інтерес гравців і заповнювати неігрових контентом перерви між виходом епізодів. А це досить затратно. Проте ряд інді-студій використовують або готові впровадити таку модель.

– Внутрішньоігрова реклама та партнерство. Ігрові компанії також отримують дохід за рахунок торгівлі особистими даними користувачів і показом реклами (банерна, повноекранна реклама або перегляд відео з винагородою, яка передбачає бонуси всередині гри). При цьому, як уже говорилося, зазвичай передбачена платна опція скасування показу рекламних роликів.

– Стягнення абонентської плати. Абонентська плата може стягуватися за ігровий час, доступ до бібліотеки ігор, ексклюзивним знижкам і новинкам за тиждень до прем'єри, власний сервер в якості послуги та ін. При цьому спостерігається гнучкість підходів до встановлення термінів дії підписочного сервісу. Так, це може бути не обов'язково річний або місячний період оплати, а пакет з 24 годин, які можна витратити в будь-який час, або, наприклад, ліміт до наступного поразки.

– Моддінг і продаж користувацького контенту. Сьогодні важливою складовою ігрових всесвітів стає призначений для користувача контент, який перетворюється в запотребований товар. Сам процес створення контенту сприймається користувачами як гра, а вбудовані інструменти для модифікування ігор (моддінга) дозволяють ефективно утримувати цільову

аудиторію. Крім того, моддінг, на думку самих розробників, є своєрідною кузницею ідей і кадрами для ігрової індустрії.

Навколо ядра, що складається з розробників ігор і видавців, розвиваються окремі підрозділи ЗМІ, спеціальні інструменти фінансів та маркетингу; з'являються спеціальні фізичні інфраструктури (інтернет-кафе для підростаючих, локації для проведення кібертурнірів і ін.); розробка ігор тепер викладається і в закладах освіти, що збільшує мабутню кількість спеціалістів, а збільшити компетентність викладачів, завжди допоможе ШІ із залученням інформаційних технологій із виявленням дезінформації [6].

Варто також врахувати впровадження технологій інтелектуального спрямування для навчального процесу, що використовуватимуться впродовж розробки ігор [7].

Крім того, зайняти свою нішу хочуть і гравці з суміжних ринків - виробники електроніки (наприклад, Apple) або інтернет-корпорації (наприклад, Google), що володіють всіма ресурсами для виведення на ринок перспективних ігрових рішень. Що стосується ключових тенденцій, що визначають розвиток глобальної ігрової індустрії, то тут, у першу чергу, необхідно відзначити нові технології: мобільні, хмарні, віртуальні, графічні, штучний інтелект та ін.

Уже протягом кількох років комп'ютерні ігри активно інтегруються в соцмережі. Також з'явився і зворотний тренд - вони самі стають своєрідними соцмережами.

Кількість людей, які цікавляться кіберспортом, зростає. Сьогодні турніри з кіберспорту вже можна порівняти з традиційними видами спорту за грошовою вартістю призу. Наприклад, призовий фонд кібертурніру Dota 2 у 2020 році оцінювався в 219 мільйонів доларів [8]. Крім того, каталізатором створення цього напрямку став кіберспорт: онлайн-трансляції гри в реальному часі, які ведуться на спеціальних платформах. Користувачі готові як грати, так і спостерігати, як грають інші, це тому, що потокове передавання забезпечує повне розуміння того, що таке комп'ютерні ігри.

З метою охоплення нових сегментів ринку і пошуку нових джерел доходу ряд учасників галузі переходить на матеріальні ринки. Зокрема, мова йде про продажі товарів з символікою бренду (футболки, браслети, чашки і т.д.), організації тематичних парків (парки Angry Birds і ін.) і т.д.

Окремо слід відзначити, що сьогодні розробники комп'ютерних ігор активно стали використовувати статистику користувача інформації. Навіть формуються спеціальні відділи для збору і обробки даних. Так, розробники трансформують свої популярні ігри в гнучкі платформи, що реагують на поведінку користувачів [9]. Використовуючи статистику, розробники змінюють складність гри, доповнюють її.

Отож, допоки ігровий світ розвивається, не потрібно відставати, натомість робити мікро-кроки аби зрозуміти всю його велич.

## **1.2 Дослідження програмного забезпечення необхідного для розробки**

Перш аніж приступати до будь-якого виду діяльності, розробнику потрібні обмеження, так зване середовище і засоби, з якими йому потрібно буде співпрацювати. В час поширеності та легкого доступу до програм на будь-який смак, є й ті, про використання яких потрібно тільки мріяти. Проте вправні користувачі ПК використовують так званий класичний спектр. А ось коли діло доходить до розробки ігор, то тут завдання ускладнюється.

Сама собою розробка ігор представляє сукупний послідовний процес з'єднання декількох творчих та технічних ланок між собою. І їхня гармонія в кінцевому результаті дає якісний продукт. Зазвичай, участь у розробці беруть декілька людей, де кожен працює над своїм вкладом. А так як даний проект виконуватиметься лиш однією особою – якість буде помітною. Але ж ціль продемонструвати кожну складову, тому потрібно визначитися із програмним забезпеченням, що посприє досягненню цієї мети.

Перед початком вибору інструментів потрібно визначити, з чим саме прийдеться мати справу. Перш за все нам потрібне середовище із об'єктами.



Місце, де розгортатиметься розробка – рушій Unity. Але коли ми говоримо про моделювання, ця програма є далекою від цієї теми. В такому випадку прийдеться вибирати спеціалізовану програму для цієї частини роботи. Їх є багато, але розглянемо одні з найпопулярніших:

- Autodesk 3ds Max;
- Blender;
- SketchUP;
- ZBrush.

Autodesk 3ds Max – це справжній продукт «все в одному». Чудовий універсальний продукт, який спрощує 3D-моделювання, візуалізацію та анімацію, створення простих і складних геометрій, дизайн функцій та текстурування. Використовується в архітектурі, дизайні, ігровій та кіноіндустрії. Користувачі можуть візуалізувати різні рівні складності та досягти реалістичного фотореалізму. Autodesk 3ds Max ефективно створює складні сцени за допомогою анімованих переходів, симуляції фізичного руху та скелетної анімації [10]. 3ds Max пропонує безліч додаткових плагінів і бібліотек (моделей, матеріалів), які дозволяють розширити функціональність програми і зробити роботу з 3D-графікою більш ефективною. 3ds Max інтегровано з іншим програмним забезпеченням Autodesk, таким як AutoCAD і Revit.

Blender – безкоштовна програма з відкритим вихідним кодом із потрібним набором інструментів для 3D-моделювання, анімації, композиції та візуалізації [11]. Це інструмент для створення 3D-графіки має багато шанувальників, оскільки він відкритий. Користувачі надають безкоштовну технічну підтримку, тож це демонструє силу ком'юніті. Завдяки об'єктивно зрозумілому інтерфейсу користувачі схиляються саме до нього.

ScetchUP та ZBrush являються програмами більш професійного рівня в галузі архітектури чи дизайну інтер'єрів, екстер'єрів будівель, скульптингу та технологіями гіперреалізму. Суто для розробки ігор вони будуть надто вимогливими щодо використання ресурсів пристрою.

Згідно обмеженого бюджету, виділеного під розробку відеогри було прийняте рішення працювати саме із безкоштовно програмою Blender. На рисунку 1.2 продемонстровано інтерфейс вибраного інструменту.

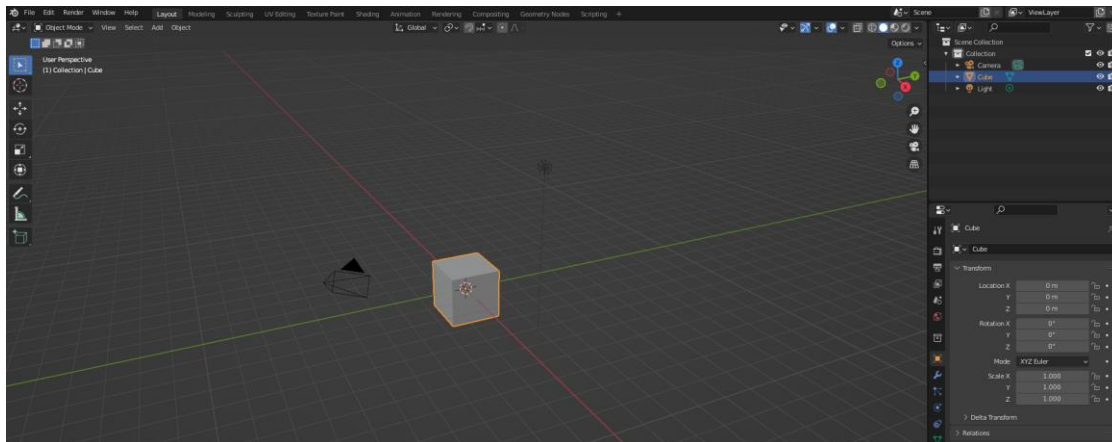


Рисунок 1.2 – Інтерфейс програми Blender

Невід’ємною частиною розробки є ще й сам дизайн об’єктів, які бачить користувач. Серед них: іконки, логотип гри, UI-складова. Із використанням Adobe Photoshop ми зможемо попрацювати над дизайн-частиною нашого проєкту [12]. Ця програма була вибрана через велику базу навчального матеріалу та простоту інтерфейсу, що повпливало на зменшення часу, потрібного для опрацювання дизайн-складової під час розробки. Приклад роботи в Photoshop наведено на рисунку 1.3.

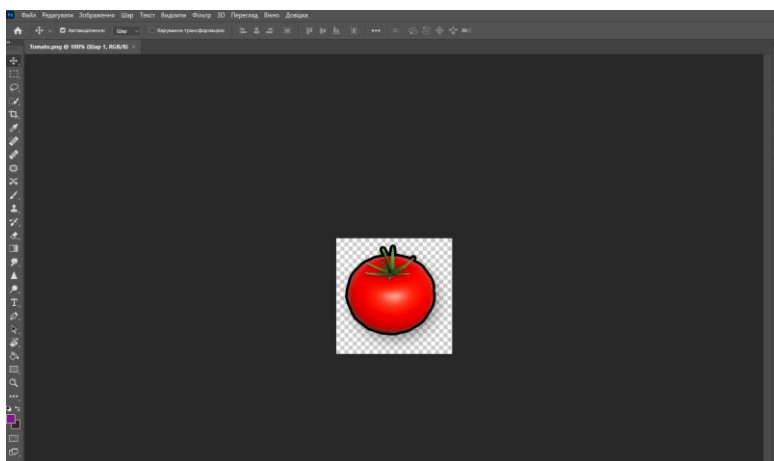


Рисунок 1.3 – Створення іконки у Photoshop

Аби кожна частина інтерфейсу користувача виглядала злагоджено, було прийнято рішення користатися програмою зовні для швидкого моделювання майбутнього відображення візуальних компонентів. В нашому випадку цим незамінним інструментом стане Figma.

Хоч вона являється програмою здебільшого для використання у розробках веб-дизайну, все ж для майбутнього проектування інтерфейсу у грі “Existence for a while” вона також годиться. Крім того, оскільки Figma є таким потужним інструментом для співпраці, існує величезна спільнота дизайнерів і розробників, які постійно створюють нові плагіни, шаблони та віджети для загального використання [13]. На рисунку 1.4 продемонстровано роботу у програмі Figma.

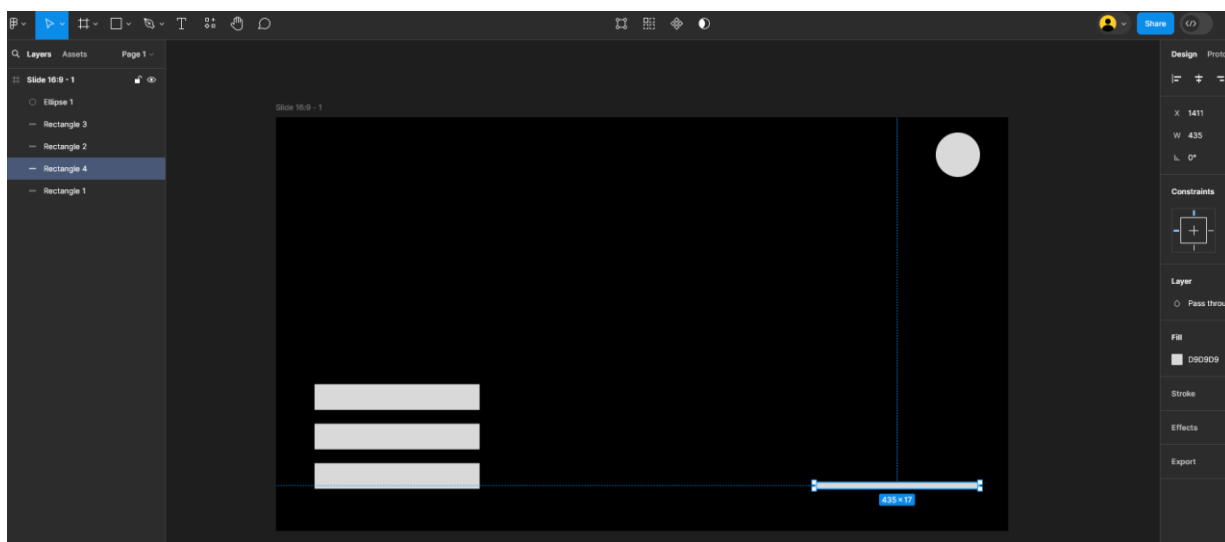


Рисунок 1.4 – Процес створення UI-дизайну у Figma

Звуковий супровід просто невід’ємна частина будь-якої гри, завдання якої привернути до себе увагу і сподобатись гравцям. Тож опрацювання цієї частини ділиться додатково ще на дві частини: спеціальні звукові ефекти навколишнього середовища, інтеракції, ходьби персонажа, сповіщень; а також музичного супроводу у самій грі.

Для виконання цієї мети було підібрано DAW FL Studio. Хоча програма все ще чітко орієнтована на створення електронної музики та звуків

«в коробці», а не на запис живих музикантів, які грають на акустичних інструментах, все ж, з допомогою неї можливо записати або створити майже будь-який аудіопроект [14]. Інтерфейс програми із її інструментами зображений на рисунку 1.5.



Рисунок 1.5 – Інтерфейс та інструменти програми FL Studio

Оглянувши список використовуваних програм уже можна сказати про глобальність GameDev індустрії. Але в проєктах із декількома розробниками список програм може доповнитись різними програмами монтажу, звукозапису, програмами для розробки сценаріїв, для дистрибуції, тощо.

### 1.3 Основний спектр інструментів при роботі із 3D рушієм Unity

Рішення вибрати саме цей рушій прийшло не одразу, адже на ринку є програми-конкуренти, що не поступаються якістю. Та найкращим доказом, чому саме Unity [15], буде його порівняння із альтернативними рушіями на ринку, порівнявши їх переваги та недоліки [16]. В таблиці 1.1 наведено порівняння ігрових рушіїв сьогодення.

Таблиця 1.1 – Порівняння популярних ігрових рушіїв

<b>Ігровий рушій</b>	<b>Фінансові витрати</b>	<b>Переваги</b>	<b>Недоліки</b>
Unity	Безкоштовний	Найбільша кількість інформації на просторах інтернету, додатковий зарібок рекламою	Посередня графіка, складність інтерфейсу, багато помилок
Godot	Безкоштовний	Оптимізований, Відкритий до багатьох мов програмування	Інтерфейс
Unreal Engine	Безкоштовний	Найякісніша 3D графіка, Хороший функціонал, Blueprints програмування, Велике ком'юніті	Легкість для новачків, ігри потребують велику к-сть ресурсів
GameMaker	Вартість 39\$	Важкість для новачків, час на розробку	Неповний функціонал, Старий дизайн
CryEngine	Безкоштовний	Якісна графіка, кросплатформеність	Велика кількість багів, відсутність підтримки

Отож, проаналізувавши нашого кандидата із конкурентами і, беручи до уваги, що апаратні ресурси обмежені, все ж рішенням буде вибрати рушій Unity для розробки відеогри. На рисунку 1.6 зображено інтерфейс рушія Unity.

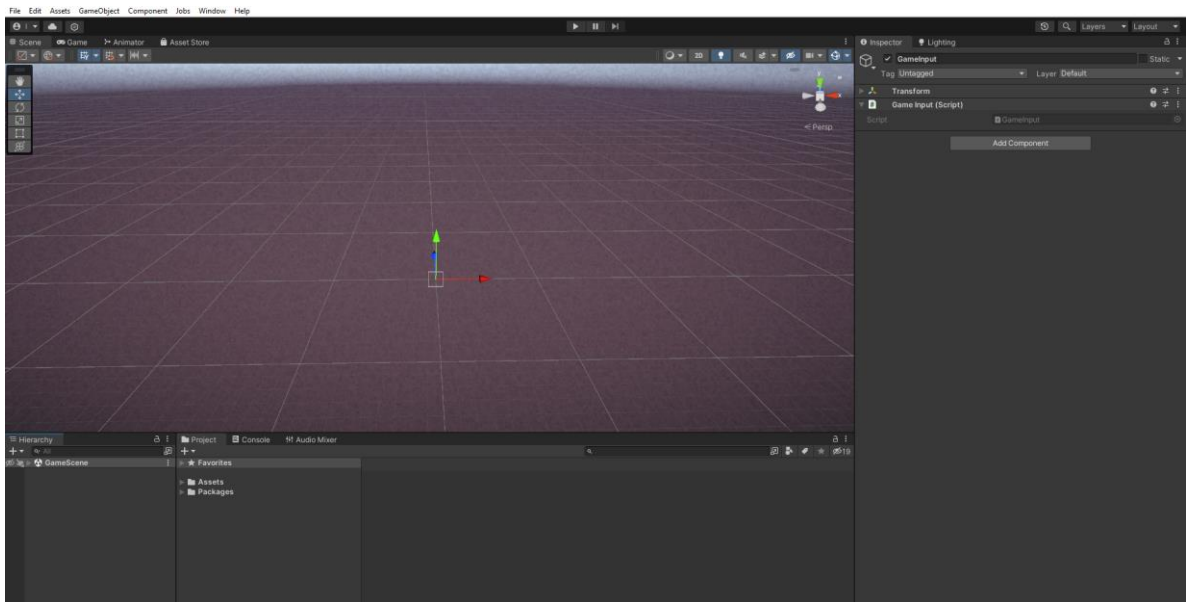


Рисунок 1.6 – Інтерфейс рушія Unity

Визначившись із інструментом середовища, настав час глянути на його функціонал [17], так як окрім кастомних налаштувань і розширень, у Unity вбудована система бібліотек і підкачок прямо із серверу, що значно спрощує та покращує процес розробки.

Серед інструментів рушія Unity [18], що використовуватимуться в проєкті, будуть:

- Анімації. Функції анімації Unity включають анімацію з можливістю перенацілювання, повний контроль над зважуванням анімації під час виконання, виклик події відтворення анімації, складні ієрархії та переходи кінцевого автомата, змішані форми для анімації обличчя тощо.

- UI-інструменти. UI Toolkit — найновіша система інтерфейсу користувача в Unity. Він створений для оптимізації продуктивності на різних платформах і базується на стандартних веб-технологіях. Є можливість використовувати UI Toolkit для створення розширень для редактора Unity, а також для створення інтерфейсу користувача для ігор і програм.

- Компонент камери. Те, що бачить камера, визначається її трансформацією та компонентом Camera. Положення трансформації визначає точку огляду, його вісь вперед (Z) визначає напрямок перегляду, а його вісь і

вісь угору (Y) визначають верхню частину екрана. Параметри компонента «Камера» визначають розмір і форму області, яка потрапляє в область перегляду. З налаштованими цими параметрами камера може відображати те, що вона «бачить» на екрані. У міру того, як GameObject рухається та обертається, відображене зображення рухається та обертається відповідно.

– Текстури. Як правило, сіткова геометрія об'єкта дає лише приблизне уявлення про форму, тоді як більшу частину дрібних деталей забезпечують текстури. Текстура — це просто стандартне растрове зображення, яке наноситься на поверхню сітки. Можна подумати про текстурне зображення так, ніби воно надруковане на гумовому аркуші, який натягнутий і прикріплений до сітки у відповідних місцях. Позиціонування текстури виконується за допомогою програмного забезпечення для 3D-моделювання, яке використовується для створення сітки.

– Пост-обробка. Unity надає низку ефектів пост-обробки та повноекранних ефектів, які можуть значно покращити зовнішній вигляд програми з невеликим часом налаштування. З'являється можливість використовувати ці ефекти для імітації фізичної камери і властивості плівки або для створення стилізованих візуальних елементів.

– Система частинок (Particle system). Система частинок моделює та рендерить багато маленьких зображень або сіток, які називаються частинками, для створення візуального ефекту. Кожна частинка в системі представляє окремий графічний елемент ефекту. Система імітує кожну частинку разом, щоб створити враження повного ефекту.

Це і був перелік основних інструментів, які використовуватимуться при розробці. Деякі менш важливі хоч і відсутні в цьому списку, проте відіграватимуть свою роль уже в майбутніх стадіях життєвого циклу створення відеогри.

## 1.4 Висновок до першого розділу

Дослідження сфери розробки відеоігор дарує розуміння важливого критерію того, що інформаційні технології не обмежуються лиш спеціалістами-розробниками програмних забезпечень. Завдяки створенню нових, умово живих світів, людство демонструє свою силу до більшого. Еволюція ігор [19] – хороший приклад розвитку продукту, який не збирається збавляти ходу.

Технології ідуть далі, даючи нам плоди у вигляді нових програм, що задовільняють наші потреби. У випадку GameDev-у, із року в рік, чим раз, тим більше програм інтегруються у велику систему, розуміння якої матиме неабияку ціну в недалекому «завтра».

Нескромний перелік інструментів Unity engine дає зрозуміти обширність свого функціоналу. Іншими словами програма ніби кричить, що в ній є усе, лиш би хто тільки нею скористався. І так станеться, бо завдяки ній розробиться ще не одна гра.



## РОЗДІЛ 2. ПРОЄКТУВАННЯ ВІДЕОГРИ У ЖАНРІ COOKING SIMULATOR

### 2.1 Дослідження cooking-симуляторів – простота і цікавість

Перш аніж приступати до виконання проєкту потрібно дійти до розуміння того, чим саме, по своїй суті, буде наш продукт. Це робиться із декількох причин. По-перше, задля кращої класифікації в позиціонуванні із іншими ринковими одиницями. По-друге, задля порівняння з аналогами одножанрових проєктів.

Розробка проєкту відбуватиметься у жанрі симулятор. Ігри-симулятори – це жанр ігор, створений для імітації дій, які ви бачите в реальному світі. Метою гри може бути навчити вас чогось. Наприклад, ви можете навчитися ловити рибу. Інші ігри-симулятори беруть участь у веденні такого бізнесу, як ферма чи тематичний парк.

Зазвичай у грі-симуляторі немає чітких цілей. Гравцям надається свобода вільно бродити та досліджувати середовище. Деякі типи ігор-симуляторів можуть включати військові ігри, симулятори рольових ігор або ігри, пов'язані з бізнесом.

Все ж, щодо майбутнього проєкту, то варто сконцентруватися на окремому виді симуляторів – було обрано cooking (кулінарний) симулятор. Кулінарні симулятори — це піджанр ігор з управління ресурсами. Незалежно від типу кулінарної гри, мета завжди полягає в тому, щоб зібрати рецепти, замовлені клієнтами протягом певного часу, щоб імітувати ресторан і догодити клієнтам [20]. Спрощені версії можуть не включати фізично рухомі предмети по кухні. Натомість вони можуть мати детальніший стаціонарний інтерфейс користувача для керування приготуванням їжі. Складніші кулінарні симулятори можуть включати вторинну логіку для придбання модернізацій на кухні, прибирання безладу, обслуговування клієнтів, декорування для отримання бонусів і підвищення якості ресторану тощо.

На рисунку 2.1 наведено загальну схему ігрового циклу для цього жанру. Петлі інших ігор у межах жанру можуть відрізнятися.



Рисунок 2.1 – Загальна схема ігрового циклу cooking симулятора

Дивлячись на просту детальну схему для розробки будь-якої гри, можна зрозуміти, що ця справа простіша, аніж можна подумати. Не потрібно придумувати колеса, коли і так все справно працює. Чи все ж ні? Якою не була б відповідь, та гарантувати можна точно, що ігри у цьому жанрі, при якісній розробці завжди будуть цікавими і не в останню чергу завдяки своїй простоті.

## 2.2 Огляд допоміжного інструменту Unity Asset Store

Маючи на меті розробку повного циклу комп'ютерної 3D-відеогри на базі рушія Unity, потрібно використати потенціал платформи на максимум. Але все ж найважливішим при розробці чи проектуванні є використання сторонніх

бібліотек чи площадок для розширення можливостей та меншої витрати часу. Отож, потрібно скористатися важливим інструментом – Unity Asset Store.

Unity Asset Store має велику бібліотеку безкоштовних і комерційних активів, створених Unity Technologies та членами спільноти [17]. Ресурси варіюються від текстур, моделей і анімацій до повних прикладів проектів, навчальних посібників і розширень. Вміст, доступний у Asset Store, дозволяє доповнити проект, гру чи програму та зменшить обсяг роботи, необхідний для створення інструментів і моделей з нуля. Активи можна придбати на веб-сайті Asset Store і додати до свого проекту за допомогою вбудованого менеджера пакетів редактора. Головну сторінку цього інструменту продемонстровано на рисунку 2.2.

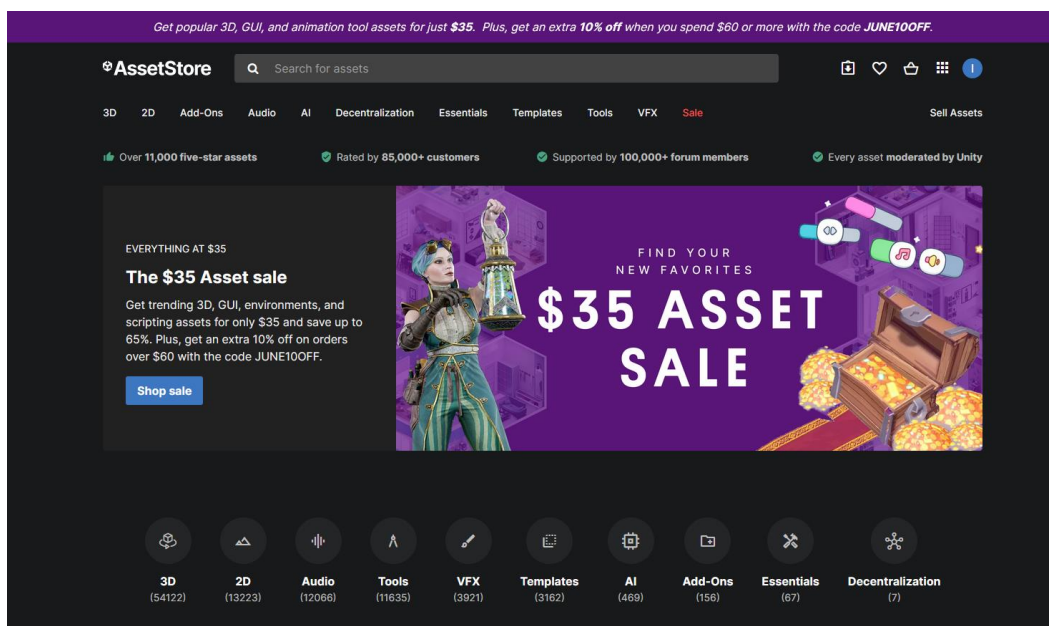


Рисунок 2.2 – Головна сторінка Unity Asset Store

Asset Store підходить у випадку, коли ресурси пристрою, на якому проводиться розробка, обмежені. Адже завдяки оцінкам та відгукам інших користувачів платформи є можливість підібрати лише оптимізовані варіанти об'єктів, в яких є потреба.

Ця платформа є лояльною до розробників окремих компонентів для ігор, що публікують свої роботи у її межах. Тож Unity Asset Store можна

використовувати і як спосіб заробітку, продаючи свої витвори потребувачим [21].

Щодо залучення цієї бібліотеки до поточного проєкту, із Unity Asset Store було використано бібліотеку SFX-звуків, через технічну нездатність синтезувати звуки у поточному середовищі розробки. На рисунку 2.3 наведено використану бібліотеку звукових ефектів Surface Sound із платформи Asset Store.

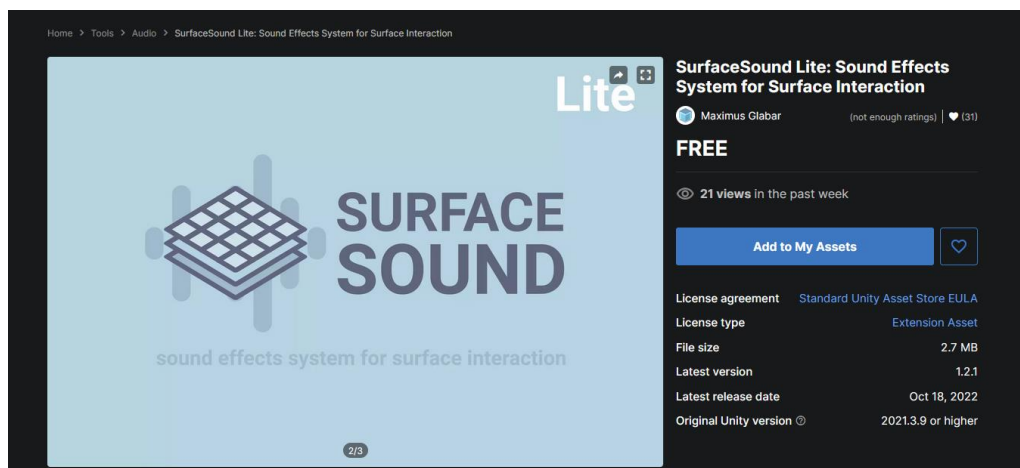


Рисунок 2.3 – Бібліотека звукових ефектів Surface Sound

Без перебільшень, Unity Asset Store це потужний інструмент для розробників, які хочуть прискорити процес розробки ігор, використовуючи готові високоякісні ресурси, додатково заощаджуючи час на інші не менш важливі речі.

### 2.3 Проєктування структури комп'ютерної гри

Розробка структури комп'ютерної гри включає багато важливих етапів, від початкового концептуального дизайну до детальної технічної реалізації [22]. Коли мова іде про ігри в жанрі симулятори, потрібно дотримуватись норм цього напрямку: казуальність, аркадність, цікавість. Сюжету, як такого, на

даному етапі розробки не передбачено, але от сам геймплей підкреслює три критерії симулятору.

Аналіз ринку демонструє [23], що попит на ігри-симулятори із роками зростатиме, залучаючи все більше потенційних гравців у сферах, де раніше не звертали уваги на їхній потенціал. “Existence for a while” – перший крок у жанрі cooking-симулятору, що підійде як і любителям ігор на реакцію, а також для часовбивання.

Дії відеогри відбуватимуться безпосередньо на кухні. Це умовний ресторан, куди приходять невидимі відвідувачі. Метою відеогри буде виконання замовлень. Замовлення у свою чергу вказуватимуть, що саме потрібно буде приготувати – кількість та тип інгредієнтів. Ігрова зона представлена: звичайними тумбами, тумбами із інгредієнтами, дошками для нарізання інгредієнтів, поверхнею для обжарювання інгредієнтів, тумбою із посудом, відром для сміття та тумбою для видачі. Зона взаємодії і в цілому весь ігровий світ обмежений тумбами. Гравця потрібно буде спішити, тож головною умовою буде додавання таймера зміни, після завершення якого гравцю виводиться інформація про виконані замовлення, а також це означає, що гру завершено.

Проект включає один рівень із початковою зоною. Проте, працюючи в ігрових рушіях по типу Unity, повноцінна відеогра не може включати лиш сцену із рівнем [24]. Тому в загальному гра складається із трьох сцен:

- Головне меню.
- Сцена завантаження.
- Ігрова сцена.

Персонажем виступає качкоподібний герой, що в кумедній анімації тинятиметься з одного кутка в інший, обмежений безжалісними тумбами, так як цього захоче його виконавець. Завданням гравця буде брати тарілку і складати на неї комбінацію продуктів задля страв, вказаних грою. Деякі продукти одразу готові, деякі потрібно додатково порізати, деякі посмажити. У останніх навіть є поріг смаження, перейдучи який вони можуть згоріти, ставши непридатними

для видачі. Замовлення можна видавати тільки на тарілці. Якщо щось пішло не так, погану комбінацію можна завжди викинути у смітник.

“Existence for a while” це тривимірна гра із виглядом камери під 40°. Таке її положення демонструє всю ігрову область та виступає зручним вікном для гравця, завдання якого якомога швидше виконувати замовлення [25].

Як і повноцінна відеогра, даний проєкт повинен мати паузу, можливість виходу під час гри. Додатково задля більшого контролю користувачем над грою було вирішено обавити вікно налаштувань із можливістю змінювати рівні гучності звуків у грі та змінювати клавіші, що відповідають за інтеракцію персонажа. Також слід враховувати різні UI-елементи, що направлятимуть гравця: картинка-тutorіал на початку гри, відлік до початку гри, таймер протягом якого триватиме гра, список поточних замовлень, таймери приготування їжі, UI-вказівник над тумбою для нарізання [26].

Загалом, навігація гравця у відеогрі представлена діаграмою наведеною на рисунку 2.4.



Рисунок 2.4 – UML-діаграма навігації у грі “Existence for a while”

Все вищезгадане буде розбавлене анімаціями, приємним дизайном, звуками та музикою аби перебування ня кухні відчувалося по-справжньому.

### 2.3.1 Проектування початкового екрану

Початковий екран або, іншими словами, головне меню – перше, що бачить гравець після запуску гри, тому, як мінімум, ця сцена має мати цікавий вигляд.

Головними об'єктами головного меню є кнопки, що переправляють гравця або безпосередньо у гру, або дають можливість закрити програму. Також не менш важливим буде розміщення логотипу даного cooking-симулятора. Аби гравець був впевнений з чим матиме справу, було прийнято рішення додати декілька кольорових версій головних персонажів, що дивитимуться прямо із екрану. Хоч гра не являє собою жахастик, ця маленька деталь може посприяти майбутнім сиквелам франшизи “Existence for a while”.

Отож, користуючись програмою Figma, було розроблено прототип майбутнього початкового екрану, який наведено на рисунку 2.5.

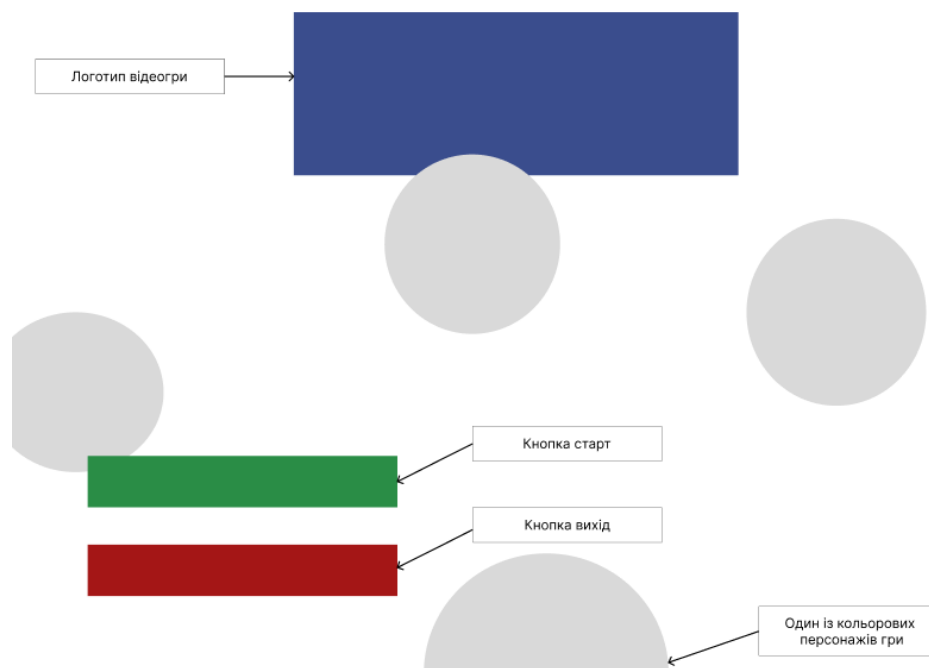


Рисунок 2.5 – Прототип початкового екрану “Existence for a while”

Таким чином, маючи макет головного меню, з’являється можливість без особливих зусиль розробити його завдяки інструментам у Unity.

### 2.3.2 Прокєтування ігрового HUD-а

Дисплей HUD складається з кількох різних елементів інтерфейсу, які накладаються на саму гру. Мета HUD — надати гравцеві важливу інформацію про поточний стан гри та гравця, не заважаючи основній дії на екрані [27].

У випадку гри “Existence for a while” елементами HUD-а виступають: список замовлень, що потрібно приготувати та таймер, що визначає тривалість гри в цілому.

Слід також враховувати, щоб елементи не мішали в процесі самої гри, тому, враховуючи жанр гри, потрібно розпланувати розташування елементів, які б не закривали ігрову область, були б зрозумілими і простими. Завдання не з важких, тому, використовуючи ту ж Figma, розробляємо макет потрібних елементів, результат роботи над яким наведено на рисунку 2.6.

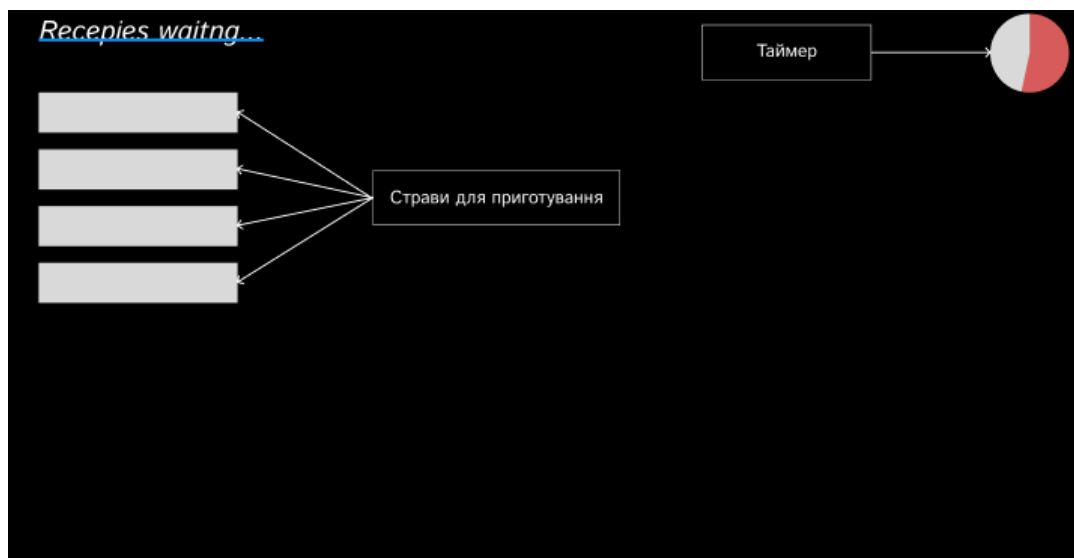


Рисунок 2.6 – Прототип HUD-а для відеогри “Existence for a while”

Гравець завжди повинен стежити за цими двома факторами, адже від них залежить кількість зроблених ним замовлень.



### 2.3.3 Прокєтування вікна налаштувань

Меню параметрів гри або вікно налаштувань дозволяє користувачам змінювати різні налаштування гри. Іноді гра має багато варіантів, іноді лише кілька. Меню параметрів, яке іноді називають меню налаштувань, пропонує той самий тип налаштування гравця [28]. Однією із фішок гри “Existence for a while” є наявність розширеного меню налаштувань із можливістю регулювання гучності та зміною кнопок управління. Зазвичай, звуки в іграх, буває дразнять вуха, також не завжди гравцям зручно грати розкладкою клавіш по замовчуванню. Виходом було створення саме цього типу меню параметрів.

Змоделюємо приблизний вигляд вікна налаштувань у Figma. Результат готового макету зображено на рисунку 2.7.

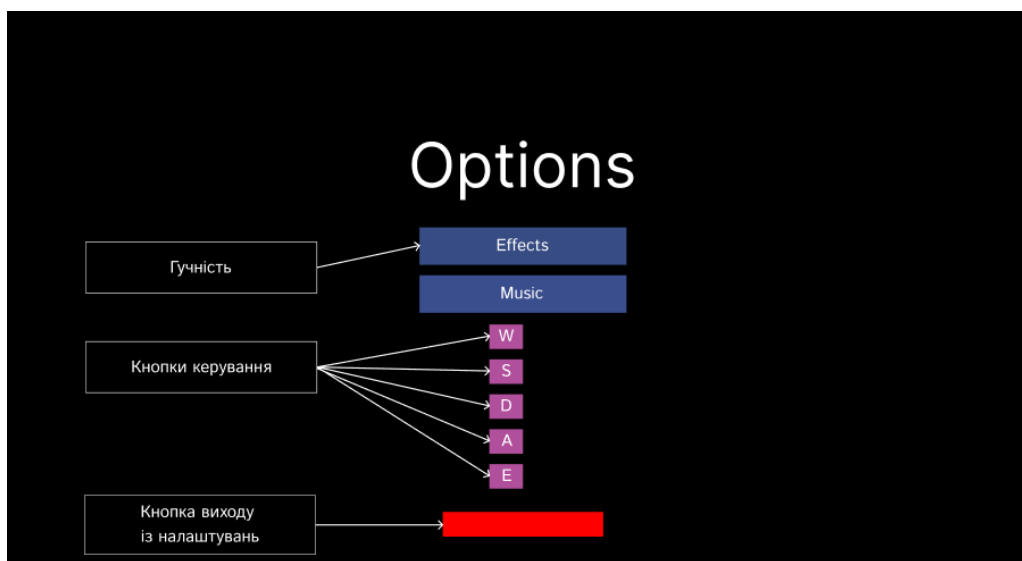


Рисунок 2.7 – Прототип вікна налаштувань для відеогри “Existence for a while”

Меню налаштувань дає великі можливості у разі ігрових доповнень, тому ним ніколи не потрібно нехтувати.

### 2.3.4 Проектування ігрової зони

Ігрова зона являє собою контроль простору між гравцем та іншою неграбельною частиною ігрового світу. “Existence for a while” матиме обмеження можливого положення гравця, тримаючи його між тумб, розташованих таким чином, щоб він із легкістю міг користуватися їх функціональними можливостями, уникаючи застрягання у текстуарах [29].

Тож рішенням було розставити майбутні тумби, що у формі дадуть прямокутник. Так як самі робочі місця, фігурами являють собою куби, зробити це буде неважко.

Важливим фактором також є розташування камери. І як раніше згадувалось кут її нахилу буде приблизно  $40^\circ$ . Цього разу для проектування ігрової зони буде використано уже 3D-середовище побудови макету ігрової зони безпосередньо у Unity engine. Результат ігрової зони, спроектованої в Unity, наведено на рисунку 2.8.

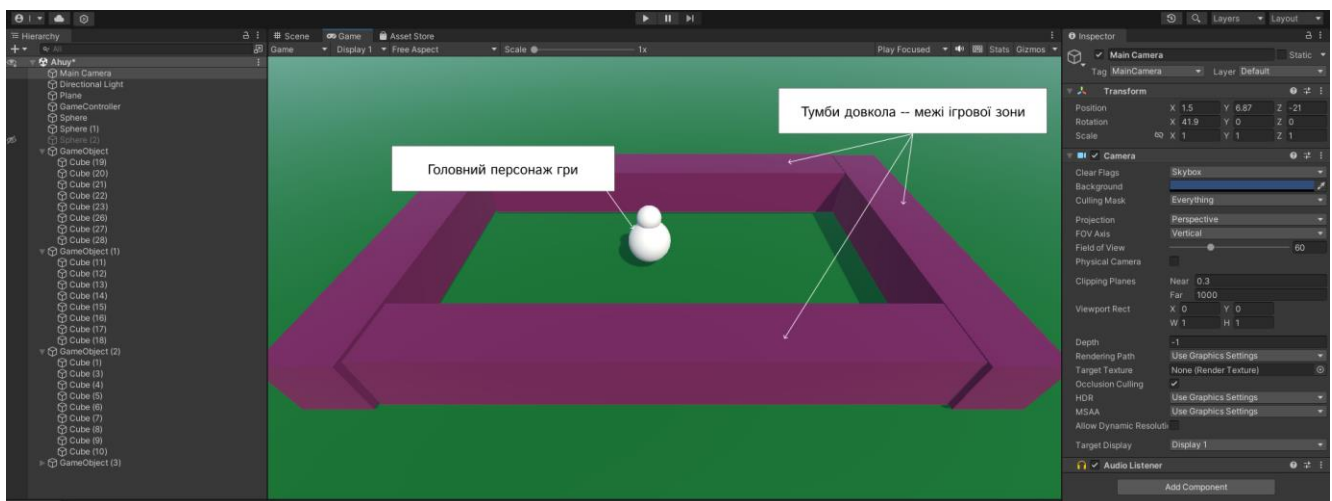


Рисунок 2.8 – Прототип ігрової зони 3D-проекту “Existence for a while” у Unity

Ігрова зона – останній етап проектування відеогри. Надалі, використовуючи цей макет, буде затрачено менше часу на розробку цієї частини проекту.

## 2.4 Висновок до другого підрозділу

У цьому розділі досліджено аспекти вибраного жанру відеогри cooking-симуляторів. Розглянуто загальну нішу жанру ігор-симуляторів. Проведено аналіз того, чому саме цей піджанр симулятор-ігор такий простий та цікавий, розглянувши сторони розвитку і побудови проекту такого типу.

Додатково розглянуто невід’ємну бібліотеку при розробці ігор Unity Asset Store, що сприяє заощадженню часу на ті елементи, на які не завжди хватає вмінь або ресурсів з боку розробника. Також ця платформа ідеально підходить для девелоперів, що самі виготовляють об’єкти для відеоігор, слугуючи їм ринком для продажів. Залучення цього потужного інструменту у межі проекту “Existence for a while” заключається у використанні готової бібліотеки звукових ефектів Surface Sound.

Спроектвано основні структурні об’єкти гри, на основі класифікації гри: попередньо проаналізувавши ринок ігор-симуляторів, описано головну мету гравця у відеогрі, із наведенням умовностей та лімітів; містяться пояснення взаємодії як персонажа із об’єктами, так і гравця із грою. Розроблено прототипи: початкового екрану, HUD-а, та вікна налаштувань із допомогою програми Figma. Ігрову зону спроектвано у середовищі рушія Unity.

## **РОЗДІЛ 3. БЕЗПОСЕРЕДНЯ РОЗРОБКА ВІДЕОГРИ ЗІ ВСІМА ЇЇ КОМПОНЕНТАМИ**

### **3.1 Розробка візуальної складової відеогри**

Розробка візуальних компонентів відеоігор є важливою частиною процесу створення гри. Він включає в себе кілька етапів: концепт-арт, створення моделей і текстур, анімацію, візуальні ефекти і інтеграцію всіх компонентів у ігровий рушій.

Графіка – це все, коли говориться про створення ігор. А в світі графіки можливості безмежні. Від яскравих і барвистих мультяшних ілюстрацій до більш докладних і реалістичних робіт – ігровий художній стиль може сподобатися користувачам і поліпшити ігровий процес. Вона дозволяє зануритися в історію з захоплюючим мистецтвом, яке переносить гравців у інші світи. Різні художні стилі гри дійсно вражають і дозволяють розробникам створювати унікальні графічні дива [30].

Серед візуальних робіт у розроблюваному проєкті є такі елементи як: іконки для позначення UI-елементів та текстур, 3D-об'єкти середовища, головний персонаж, анімації ходьби, готування, нарізання, екземпляр шейдеру, рух камери, світлокорекція та пост-обробка.

#### **3.1.1 Створення 2D-спрайтів майбутніх іконок**

Проєкт “Existence for a while”, що розробляється у жанрі cooking-симулятору, містить елементи, що інформуватимуть гравця про дії, які йому слід виконати в приготуванні замовлень. Замовлення це суміш окремих інгредієнтів, що в окремих комбінаціях складають страви. І саме тип інгредієнтів нас цікавить. Задля роботи над візуальною частиною інгредієнтів у Unity, потрібно зробити їхні 2D-спрайти. Серед інгредієнтів на кухні потрібні: хліб, капуста, м'ясо, сир, томат.

Для розробки текстур та спрайтів об'єктів для проєкту було обрано програму Adobe Photoshop. На рисунку 3.1 наведено роботу над створенням спрайту для інгредієнту “CheeseBlock”.

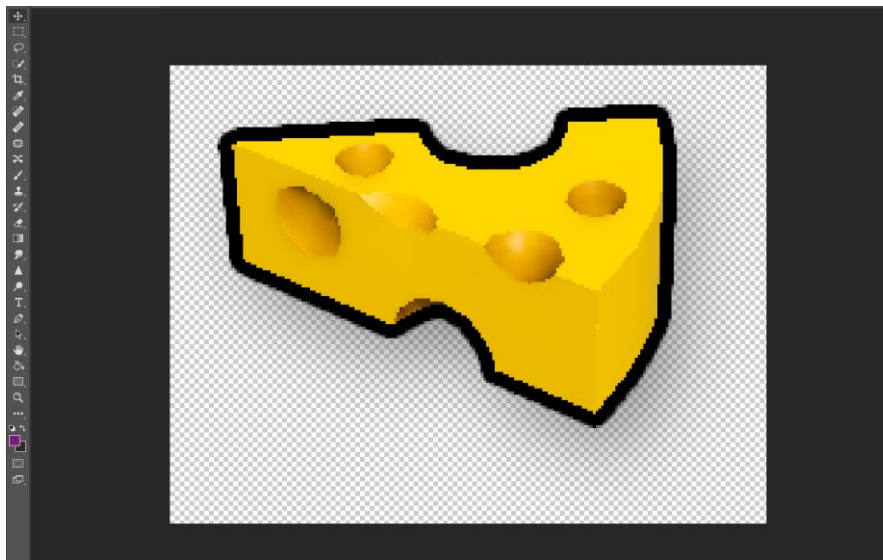


Рисунок 3.1 – Спрайт для інгредієнту “CheeseBlock” розроблений у Photoshop

При створенні спрайтів використовувалися пензлі, заливки та градієнти із інструменталу Adobe Photoshop.

По аналогії до об'єкту “CheeseBlock” створено об'єкти “Bread”, “Tomato”, “Cabbage” та “Meat”. Також розроблені альтернативні версії продуктів після готування: “TomatoSliced”, “CheeseBlockSliced”, “MeatPattyCooked”, “MeatPattyBurned”, “CabbageSliced”. Також для UI-позначення тарілок розроблено спрайт “Plate”. Всі 2D-спрайти збережено у форматі PNG а згодом імпортовано до Unity. Спрайти об'єктів всередині рушія продемонстровано на рисунку 3.2.



Рисунок 3.2 – Спрайти імпортованих об'єктів у середовищі Unity

Також потрібно створити об'єкти, що вказуватимуть на прийняття або відхилення замовлення, а також значок попередження, все у тому ж Photoshop.

Готові спрайти інформативних значків зображено на рисунку 3.3.



Рисунок 3.3 – Спрайти інформативних значків у середовищі Unity

Важливою частиною під час створення відеогри є її логотип. Логотип є важливим елементом ігрової унікальності, оскільки це перший візуальний контакт між гравцем і грою. Вдалий логотип допоможе відрізнити гру від конкурентів, створить впізнаваність і справить перші враження [31].

Згідно тематики гри у жанрі cooking-симулятора було розроблене лого зображене на рисунку 3.4.

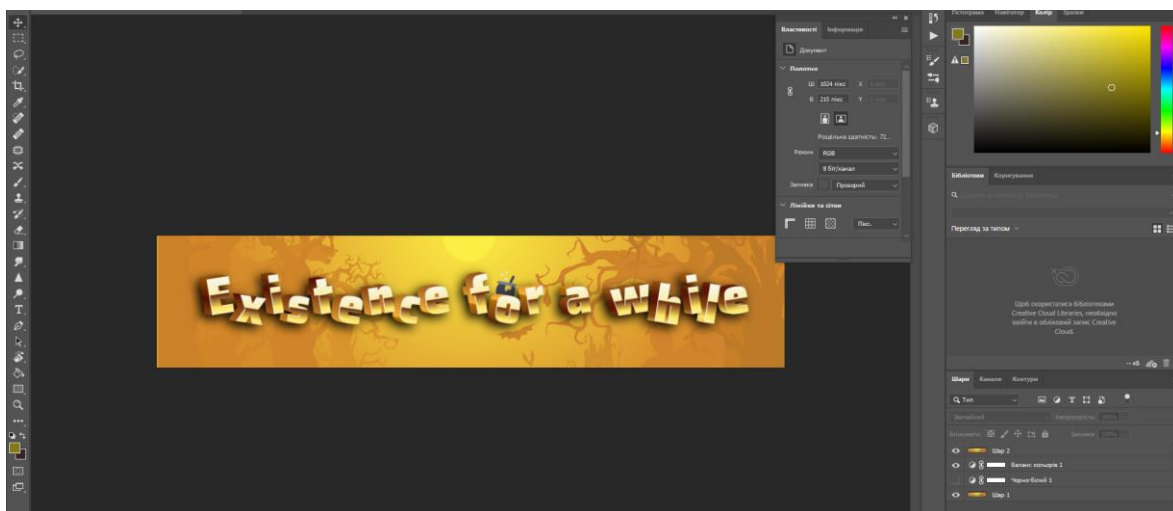


Рисунок 3.4 – Логотип відеогри “Existence for a while” розроблений у Photoshop

Додатково, для подачі інформації про ціль гри гравцеві пропонується міні-туторіал. Фінальний вигляд туторіалу поданий у додатку А. Його прототип перед розробкою, реалізований у Figma, поданий на рисунку 3.5.



Таким ж чином розроблені тумби: “ClearCounter”, “ContainerCounter”, “DeliveryCounter”, “PlateCounter”, “TrashCounter” (смітник) та “CuttingCounter”. Додатково розроблено 3D-об’єкти інших елементів та імпортовано у проєкт. Для імпортування тривимірних об’єктів у Unity потрібно експортувати готові моделі із Blender у форматі FBX. 3D-модель розробленого імпортованого у рушій інгредієнту томату наведено на рисунку 3.7.

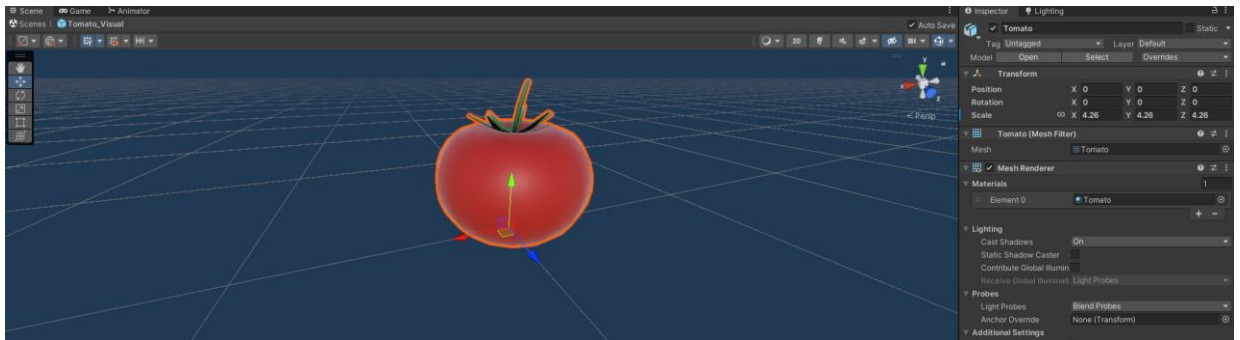


Рисунок 3.7 – 3D-модель імпортованого томату у Unity

Для даного проєкту були підібрані нескладні варіанти 3D-об’єктів середовища та взаємодії, натомість аби докласти зусиль і до інших не менш важливих елементів гри.

Всі елементи середовища та взаємодії використовуються у грі у вигляді префабів. Prefab – це збірний об’єкт, який є шаблоном GameObject. Від нього можна створювати копії, які називаються екземплярами. Коли змінити параметри префаб-об’єкта, зміняться і всі його екземпляри. Для побудови ігрової зони, було використано лиш префаби тумб.

### 3.1.3 Анімація головного персонажа та об’єктів у грі

Анімація відеоігор стала ключовим елементом у наданні гравцям захоплюючого досвіду. Прорив технологій та програмного забезпечення дозволили розробникам ігор створювати дуже складні та візуально вражаючі анімації. Анімація кожного елемента, хоч це гострі відчуття від захоплюючої



погоні або нестерпний смуток через втрату персонажа, дозволяє дизайнеру викликати емоції у гравця. Анімація не тільки покращує ігровий процес, але й є важливим інструментом для розповіді у відеоіграх та розвитку персонажів. Природно, ігрова анімація стала одним з ключових елементів успіху сучасних відеоігор, якими ми всі насолоджуємося [32].

Серед анімацій в проекті є декілька екземплярів: анімація головного персонажу, анімація предмету в «руках», анімація нарізання інгредієнтів, анімація готування на сковороді.

Анімація головного персонажа представлена двома діями: анімація бездіяльності та анімація ходьби. У Unity ці дві умовності розподіляються завдяки інструменту Animator Controller. Вказавши потрібні параметри та логічні передумови, потрібно створити стани “Idle” для бездіяльності та “Walk” для ходьби. Стани у свою чергу вміщують зміну елементів, згідно таймінгів, які визначає аніматор. Процес анімації в стані – Animation clip – залежить від умови, яку поставить програміст. Умовою у випадку розроблюваної відеогри є boolean значення “Is walking”, що змінюється в залежності від того, чи нажимає гравець клавіші ходьби за умови, що інші умови не порушуються.

Процес виконання анімації у Animator Controller поданий на рисунку 3.8.

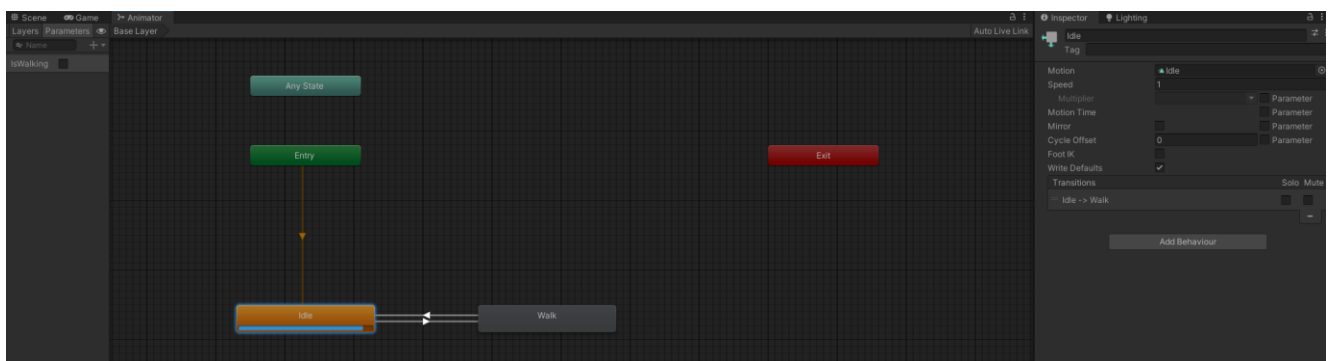


Рисунок 3.8 – Процес анімації бездіяльності у Animator Controller

Окрім головного персонажу, деякі об’єкти у грі також мають частинку життя. Обрані об’єкти в руках – левітують вверх-вниз. Фасувальні тумби мають анімацію відкривання/закривання. Тумба для нарізання містить всередині

анімацію змаху ножа. Під час готування інгредієнту “Meat” використана технологія Particle System для додавання бризків від смаження. Анімація смаження, зроблена із допомогою Particle System продемонстрована на рисунку 3.9.

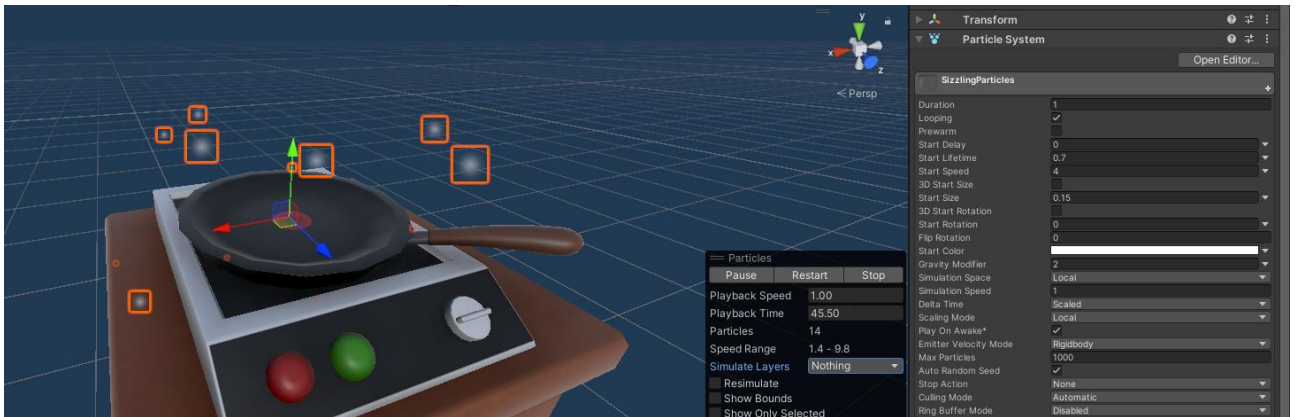


Рисунок 3.9 – Анімація смаження із використанням технології Particle System

Ця ж технологія застосована і при пересуванні головного персонажу, «ноги» якого лишать пилуку після себе.

Частинами анімацій також будуть UI-елементи прогресу, попередження про готовність продукту та вікна успішного чи неуспішного виконання замовлення та таймер тривалості гри. Приклади цих дрібних, але важливих візуальних елементів із нотками анімації продемонстровано на рисунку 3.10

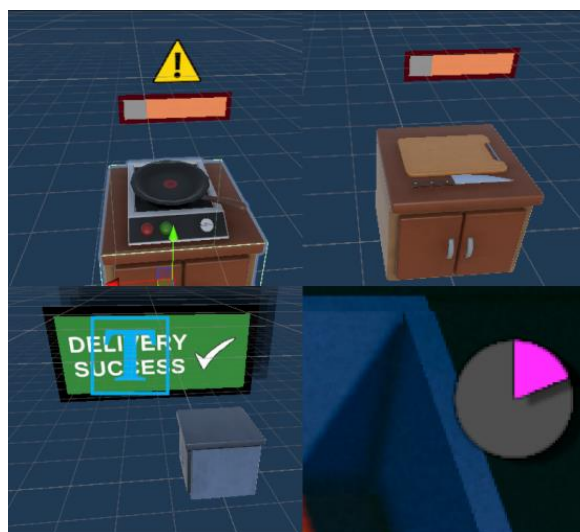


Рисунок 3.10 – Об’єкти UI-анімацій в проєкті “Existence for a while”

Розробивши UI-компоненти проєкту “Existence for a while”, продовжимо розробку елементів інтерфейсу.

### 3.1.4 Розробка шейдеру для анімації Delivery Counter-a

Комп’ютерна графіка, особливо 3D-графіка, використовує різні типи інформації для створення візуальних елементів, таких як сітки, текстури та світло. Ця інформація передається на графічне обладнання, яке обробляє зображення та відображає його на екрані. Візуалізація — це те, що програмісти називають процесом створення зображення, а шейдери — це короткі програми, які відображають графічні дані. Іншими словами, шейдер — це програма, яка приймає сітку, текстуру тощо. як вхід і створює зображення як вихід.

“DeliveryCunter” це саме той об’єкт, поверхню якого було використано для створення шейдеру – інфацію стрілки, що інтуїтивно вказуватиме гравцю на пункт куди потрібно здавати замовлення.

Для розробки шейдера, потрібно скористатися інструментом Shader Graphs, застосувавши налаштування та навички візуального програмування, продемонстровані на рисунку 3.11.

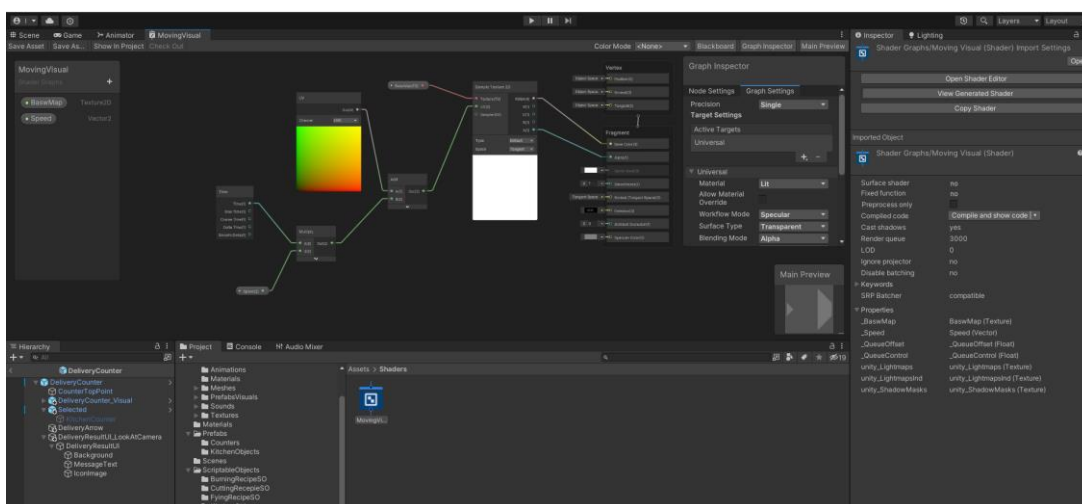


Рисунок 3.11 – Налаштування для шейдеру “DeliveryCounter”

Як результат, було отримано стрілку, що рухається вздовж поверхні тумби для видачі. Тепер це частина текстури самої тумби. На рисунку 3.12 продемонстровано тумбу доставки до та після накладання на неї шейдера.

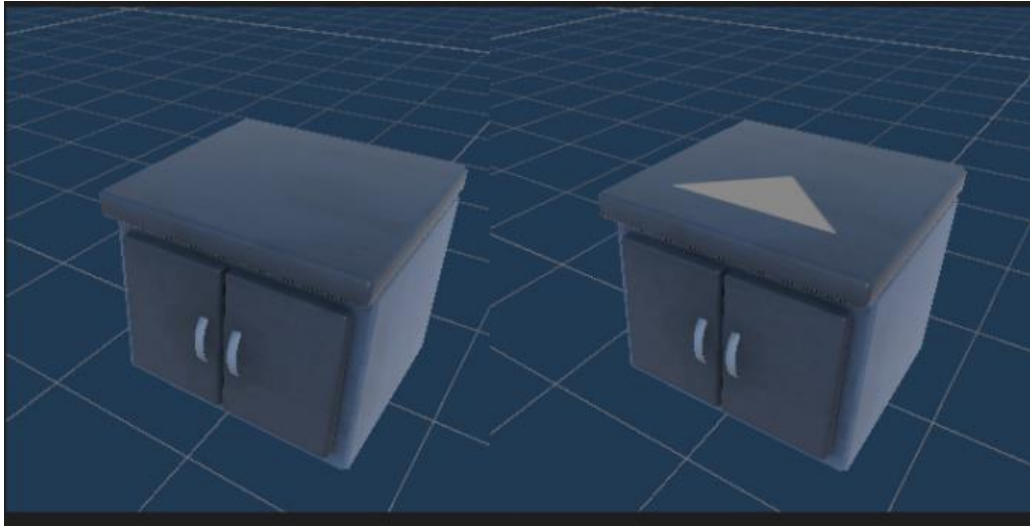


Рисунок 3.12 – “DeliveryCounter” до та після накладання на нього шейдера

Таким чином, описано всі головні анімації проєкту, результат яких не розчарує.

### **3.1.5 Налаштування камери, світлокорекція, пост-обробка**

Рух камери дозволяє експериментувати із динамічністю відеоігор, а також додати настрою до подій, що розгортаються перед гравцем. Саме завдяки камері той, що грає, має змогу споглядати все чудо, створене розробником. Але перш ніж просто налаштувати стокову камеру Unity, варто звернути увагу на розширення, які нам пропонує сам рушій.

Package Manager вміщує широкий вибір ресурсів, від текстур, моделей і анімацій до цілих прикладів проєктів, навчальних посібників і розширень редактора, а також дозволяє взаємодіяти із раніше згаданим Unity Asset Store.

Отже, коли мова йде про камеру в іграх, потрібно звертатися до Cinemachine – набору модулів для роботи з камерою Unity. Cinemachine

вирішує складну математику та логіку відстеження цілей, композиції, змішування та нарізки між кадрами. Його розроблено, щоб значно зменшити кількість трудомістких ручних маніпуляцій і переглядів сценаріїв, які мають місце під час розробки.

Користуючись цим інструментом, потрібно надати декілька налаштувань аби додати їй похитувань, фокусу та вдалого кута огляду для майбутнього гравця. Саме ця конфігураційна система камери та повноцінна розстановка об'єктів по карті ігрової зони подана на рисунку 3.13.

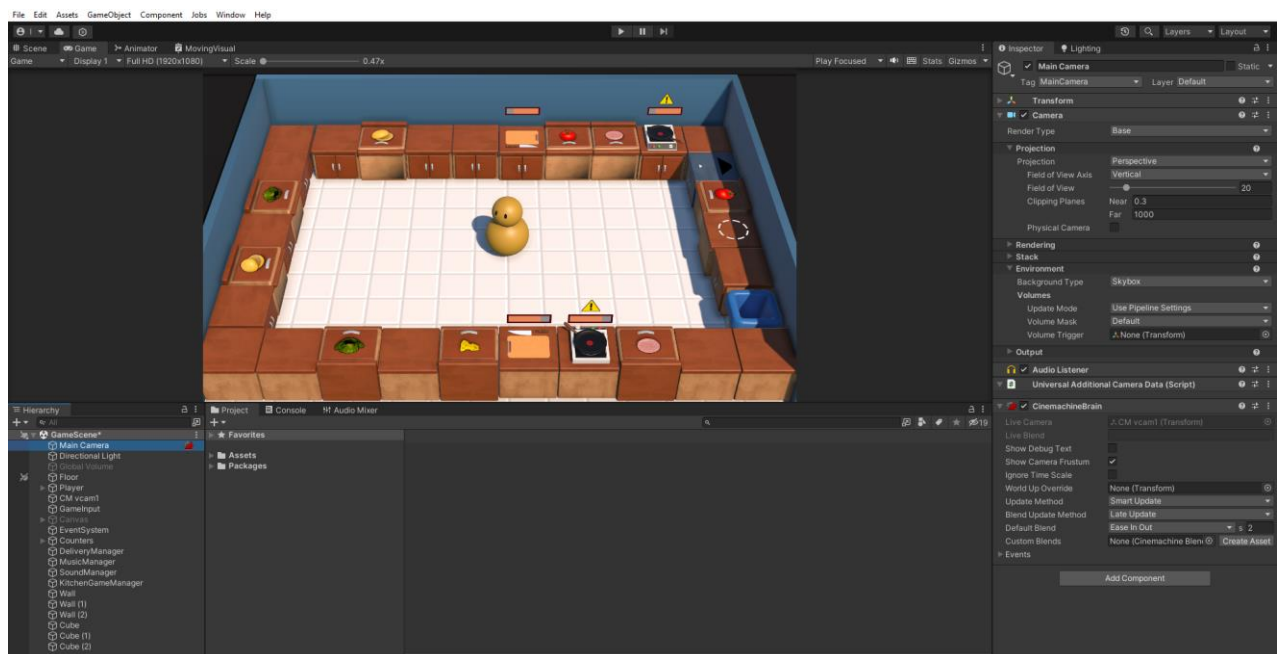


Рисунок 3.13 – Налаштування та вигляд камери Cinemachine у межах зони гри

Як можна поспостерігати, візуальна якість гри демонструє сірість кольорів і транслює сум. Тому варто погратися із корекцією світла та навичками пост-обробки. Використавши параметри: Bloom, Color Adjustment, Film Grain, Lift Gamma Gain, Tonemapping та Vignette; та доклавши часу до маніпуляцій із їхніми параметрами отримаємо зовсім іншу картинку, наведену на рисунку 3.14.

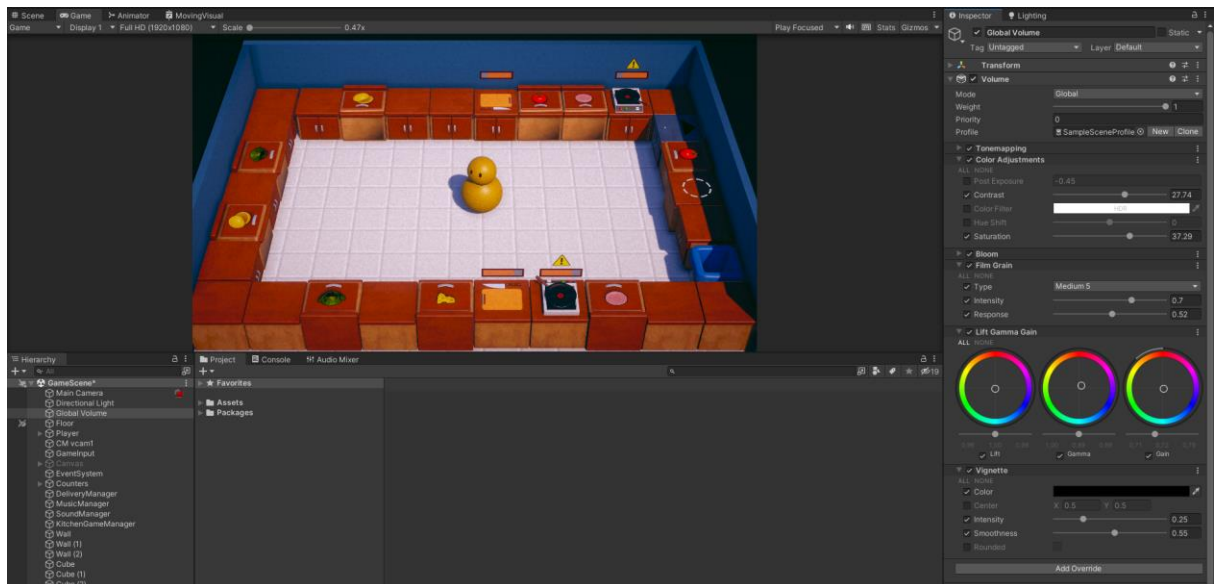


Рисунок 3.14 – Вигляд ігрової зони після ефектів світлорекорекції та пост-обробки

Виглядає значно краще, хоч це лише базові налаштування кольору. Таким чином, демонстрація компонентів камери і кольору як частин візуального доповнення показує розширені можливості інструментів як для розробника, так і для гравця.

## 3.2 Розробка логіки та взаємодії гравця із предметами довкілля

Маючи розроблені компоненти потрібно упорядкувати їхню дію та логіку використовуючи знання із програмування. Unity engine ідеально поєднується із мовою програмування C#, використовуючи методи ООП та вбудованих класів MonoBehavior. Об'єкти, екземпляри, префаби, спрайти, анімації чітко потрібно регулювати скриптами.

### 3.2.1 Програмування головного персонажа. Тестування рухів у ігровому середовищі

Найпершими та найголовнішими скриптами є введення рухів персонажеві. Для цього потрібно підв'язати інструмент Unity Input Actions до головного персонажа, перед тим вказавши які клавіші, за що відповідають. Призначення управління у Input Actions вказано на рисунку 3.15.

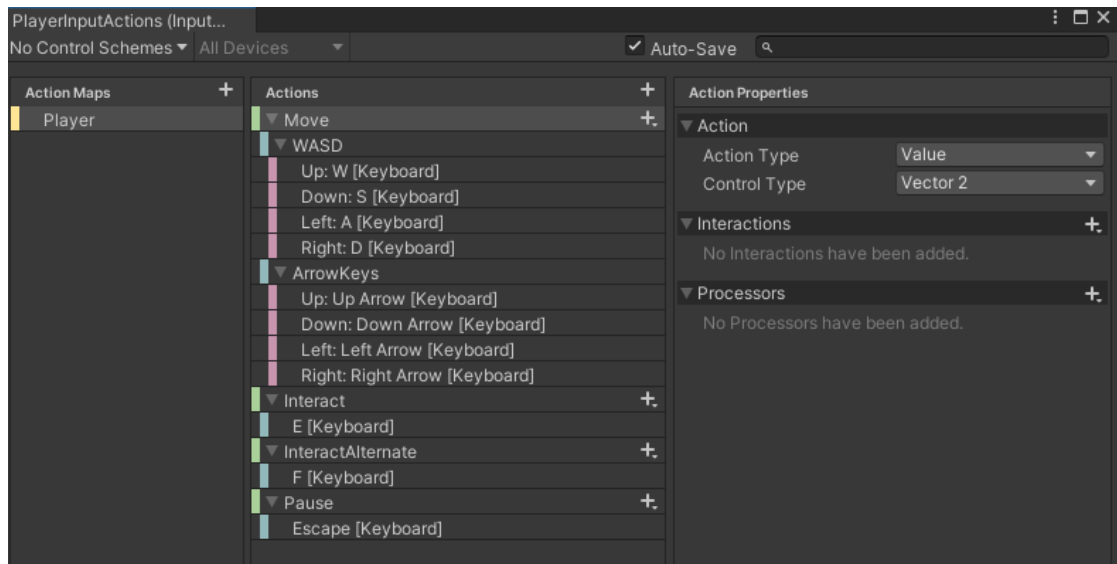


Рисунок 3.15 – Призначення клавіш для управління персонажем

Тепер за допомогою скрипту “PlayerInput” прописуємо логіку ходьби головному персонажеві, не забуваючи прикріпити до нього анімації зі скриптом “PlayerAnimator”. На рисунку 3.16 продемонстровано переміщення персонажа по мапі, включно із анімаціями.

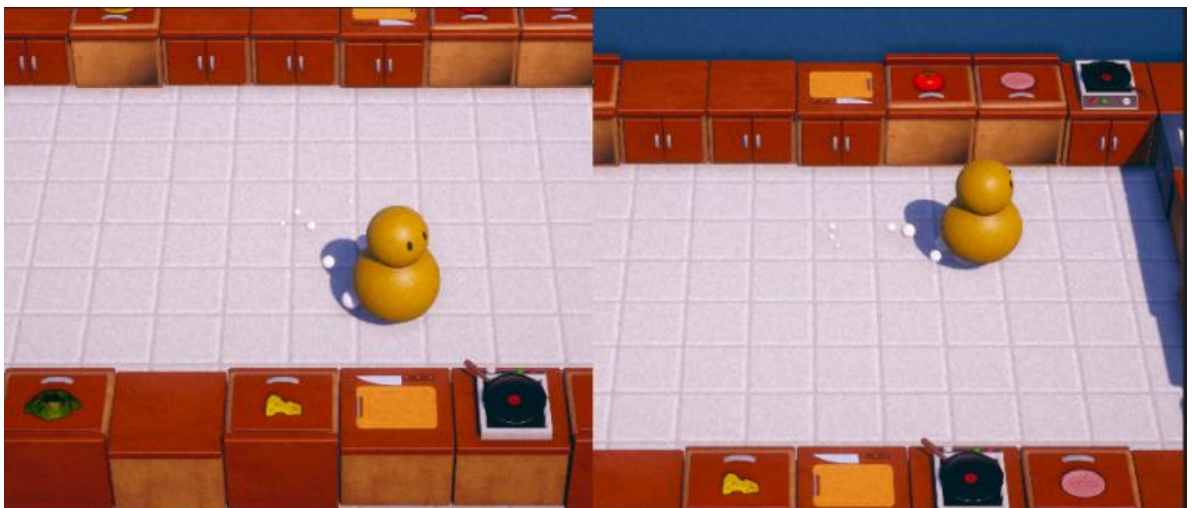


Рисунок 3.16 – Переміщення та анімація ходьби персонажа

Створюємо “EmptyObject” напроти персонажа, в місці якого буде знаходитися обраний предмет.

Зазвичай при розробці рухів персонажеві, розробники стикаються із проблемами проходження крізь текстури, випадання крізь мапу, які пов’язані із некоректним написанням коду для вектора напрямку руху. Використовуючи функцію “HandleMovement” (див. лістинг 3.1), проходження персонажа крізь текстури буде неможливою.

### Лістинг 1.1 – Програмний код рухів персонажа “HandleMovement”

```
private void HandleMovement() {
    Vector2 inputVector = gameInput.GetMovementVectorNormalized();

    Vector3 moveDir = new Vector3(inputVector.x, 0f,
inputVector.y);

    float moveDistance = moveSpeed * Time.deltaTime;
    float playerRadius = .7f;
    float playerHeight = 2f;
    bool canMove = !Physics.CapsuleCast(transform.position,
transform.position + Vector3.up * playerHeight, playerRadius,
moveDir, moveDistance);

    if (!canMove) {
        // Cannot move towards moveDir

        // Attempt only X movement
        Vector3 moveDirX = new Vector3(moveDir.x, 0,
0).normalized;
        canMove = moveDir.x != 0 &&
!Physics.CapsuleCast(transform.position, transform.position +
Vector3.up * playerHeight, playerRadius, moveDirX, moveDistance);

        if (canMove) {
            // Can move only on the X
            moveDir = moveDirX;
        }
    }
    else {
        // Cannot move only on the X

        // Attempt only Z movement
        Vector3 moveDirZ = new Vector3(0, 0,
moveDir.z).normalized;
```



```

        canMove = moveDir.z != 0 &&
!Physics.CapsuleCast(transform.position, transform.position +
Vector3.up * playerHeight, playerRadius, moveDirZ, moveDistance);

        if (canMove) {
            // Can move only on the Z
            moveDir = moveDirZ;
        }
        else {
            // Cannot move in any direction
        }
    }
}

if (canMove) {
    transform.position += moveDir * moveDistance;
}

isWalking = moveDir != Vector3.zero;

float rotateSpeed = 10f;
transform.forward = Vector3.Slerp(transform.forward, moveDir,
Time.deltaTime * rotateSpeed);
}

```

Результат тестування рухів персонажа поруч об'єкти зображений на рисунку 3.17.



Рисунок 3.17 – Тестування рухів персонажа поруч із об'єктами

Основна розробка маніпуляції над персонажем завершена. Залишились тільки звукові ефекти.

### 3.2.2 Розробка логіки для тумб

Префаби тумб із якими потрібно працювати також вважаються 3D-об'єктами, та кожна із них виконує свою функцію.

Почати варто із звичайного “ClearCounter”, що виконує функцію поверхні, на яку можна просто покласти будь-який об'єкт із рук персонажа. Тому у скрипті “ClearCounter” потрібно прописати умовності, при якій можна покласти предмет та взяти його.

По тій ж аналогії до об'єкта “ContainerCounter” прописуємо функції для отримання продуктів при взаємодії із ним у скрипті “ContainerCounter”. Для тумби “CuttingCounter” визначаємо умови використання її для нарізання. Для “PlateCounter” розробляємо систему спавну тарілок, які можна брати для фасування замовлення. Для “TrashCounter” визначаємо механізм видалення об'єктів із «рук» персонажа при взаємодії. “StoveCounter” варто взяти до уваги, адже для його об'єкта варто прописати логіку смаження, яка матиме 3 стани: процес смаження сирого інгредієнту, процес смаження готового інгредієнту, видача пересмаженого інгредієнту. Та заключний “DeliveryCounter” наділено логікою прийняття замовлення, яке розфасоване на тарілці. Саму тарілку ця тумба приймає також, але це лиш безцільний рух гравця. Скрипти до кожної із тумб мають аналогічну із ними назву.

### 3.2.3 Подання інгредієнтів у вигляді скриптових об'єктів

ScriptableObjects – це контейнер даних, який можна використовувати для збереження великих обсягів даних незалежно від екземплярів класу. Одним із основних випадків використання ScriptableObjects є зменшення використання пам'яті проекту шляхом уникнення копій значень. Кожного разу, коли створюється префаб, він отримуватиме власну копію цих даних. Замість того, щоб використовувати цей метод і зберігати дубльовані дані, було прийнято рішення використовувати ScriptableObject для зберігання даних і потім доступу

до них за посиланням з усіх Prefabs. Це означає, що в пам'яті є лиш одна копія даних.

Такими даними у розроблюваній відеогрі виступають об'єкти інгредієнтів та тарілки, саме їх можна буде брати та проводити над ними маніпуляції: нарізання, смаження, фасування. На рисунку 3.18 наведено ScriptableObject об'єктів для інгредієнтів.

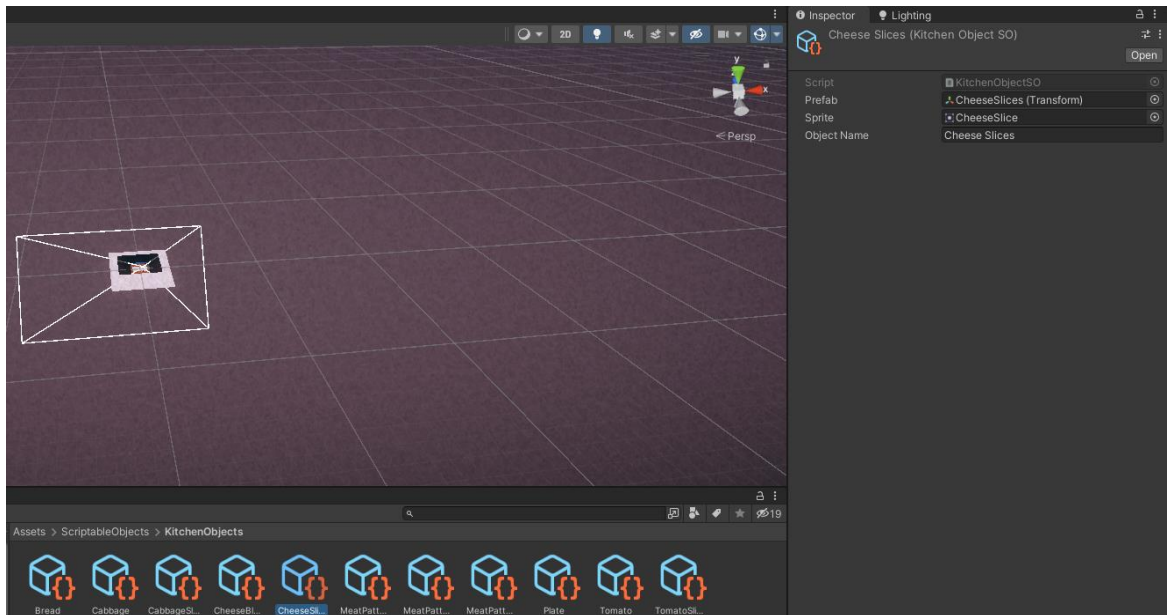


Рисунок 3.18 – ScriptableObject об'єктів для інгредієнтів

Використовуючи функції ScriptableObject, з'являється можливість прооптимізувати проєкт належним чином.

### 3.2.4 Розробка UI-компонентів у грі

Маючи прототипи UI-компонентів проєкту, потрібно переходити до їх розробки. Перш за все потрібно додати UI-показники процесу нарізання інгредієнтів для "CuttingCounter" та показник процесу готовності для "StoveCounter". Показники прогресу тумб наведено на рисунку 3.19.



Рисунок 3.19 – Показники прогресу тумби для нарізання та тумби для смаження

Також потрібно розробити початковий відлік до початку гри, щоб притримати нетерплячого гравця, блокуючи йому взаємодію із об'єктами. На рисунку 3.20 наведено таймер початку відліку гри.

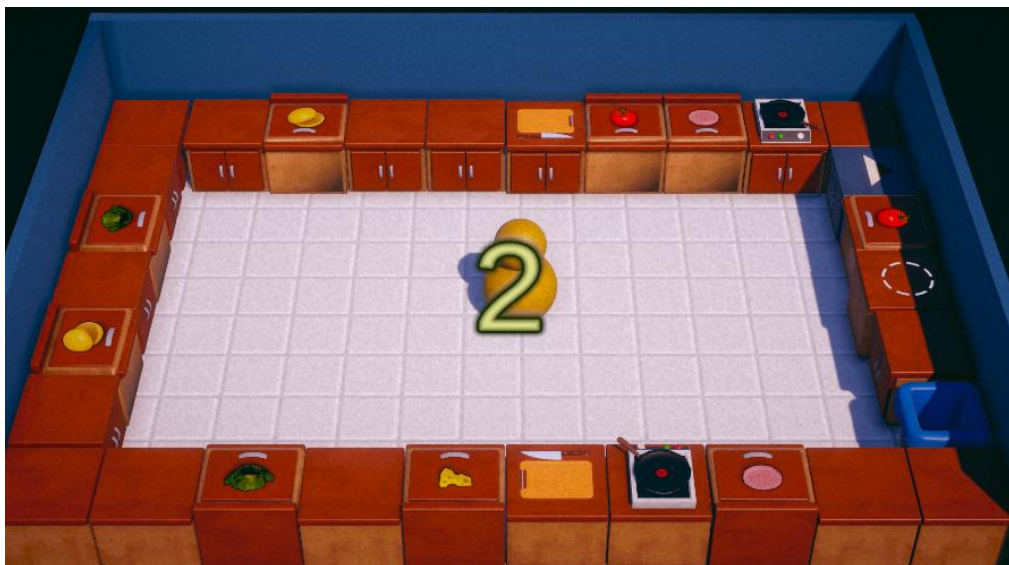


Рисунок 3.20 – Таймер відліку до початку гри

По прикладу прототипу створюємо таймер тривалості гри. Також розробляємо вікно паузи та перед грою прикріплюємо картинку-тutorіал (див. додаток А).

### 3.2.5 Створення сцен головного меню та завантаження. Вікно паузи

Перше, що повинен бачити гравець, відкриваючи файл гри – головне меню. Отож, маючи прототип, розробляємо сцену головного меню. Готова версія головного меню зображена на рисунку 3.21.



Рисунок 3.21 – Головне меню відеогри “Existence for a while”

Між головним меню та самою грою повинна міститися сцена завантаження, задля коректного підвантаження всіх елементів наступної сцени. Сценарій “LoadScene” повинен запуснитися лиш при нажиманні кнопки «Play» у головному меню. Нажавши кнопку «Quit», гравець покине гру. Вікно “LoadScene” реалізовано через інструмент Unity “EventSystem” та скриптом “LoaderComeback”.

Через невелику кількість об’єктів у проєкті, вікно завантаження з’являється лиш на долю секунди.

### 3.3 Розробка механізму замовлень – головна ціль гри

Ціль гравця у відеогрі – виконувати замовлення. Замовлення реалізовані UI-компонентом “DeliveryManager”, що інформує гравця який тип замовлення йому виконувати. Цей компонент містить базу інгредієнтів, що складаються і цілий окремий об’єкт. Завдання гравця: зібрати сет із інгредієнтів, вказаних у

“DeliveryManager”. Видача замовлень “DeliveryManager” продемонстрована на рисунку 3.22.



Рисунок 3.22 – Видача замовлень “DeliveryManager”

Скрипт аналізує чи містить замовлення відповідні інгредієнти перед підтвердженням видачі на “DeliveryCounter” і в залежності від результату або підтверджує замовлення гравця, нараховуючи йому бали, або приймає замовлення видаючи повідомлення про неправильно зроблене замовлення. Результати того чи замовлення правильне чи ні продемонстровано на рисунку 3.23.

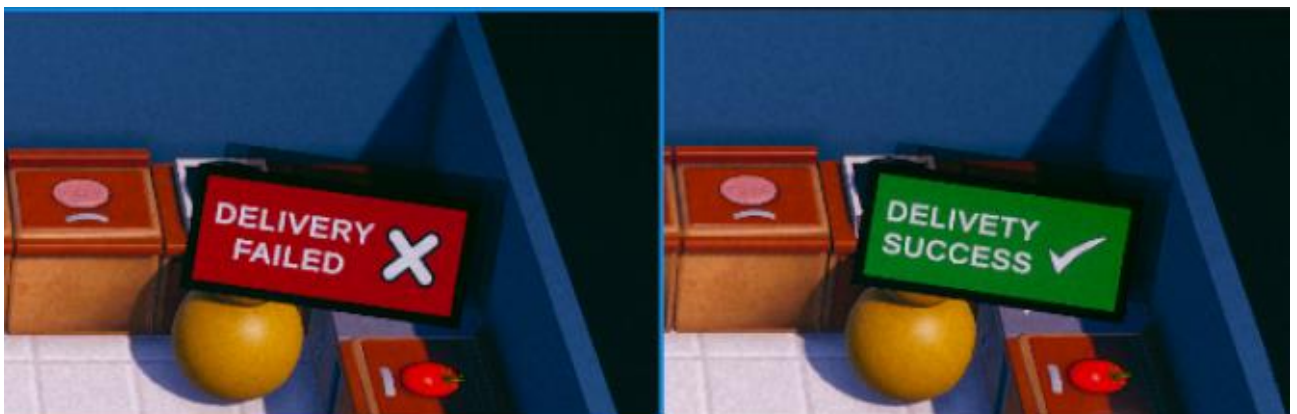


Рисунок 3.23 – Прийняття замовлення тумбою “DeliveryCounter”

### 3.3.1 Тестування валідності виконання замовлення

Під час тестування механізму замовлень критичних «багів» виявлено не було. Єдиною проблемою залишається лише одноразовий показ успіху вдалого замовлення “DeliveryCounter”, але ця проблема легко вирішиться додаванням звукового супроводу, що вказуватиме на успіх або відмову від замовлення.

В кінці гри після завершення таймера всі успішно виконані замовлення додаються і UI-елемент закінчення гри інформує гравця про їх загальну кількість. На рисунку 3.24 продемонстровано вікно “GameOver” із результатами виконаних замовлень.

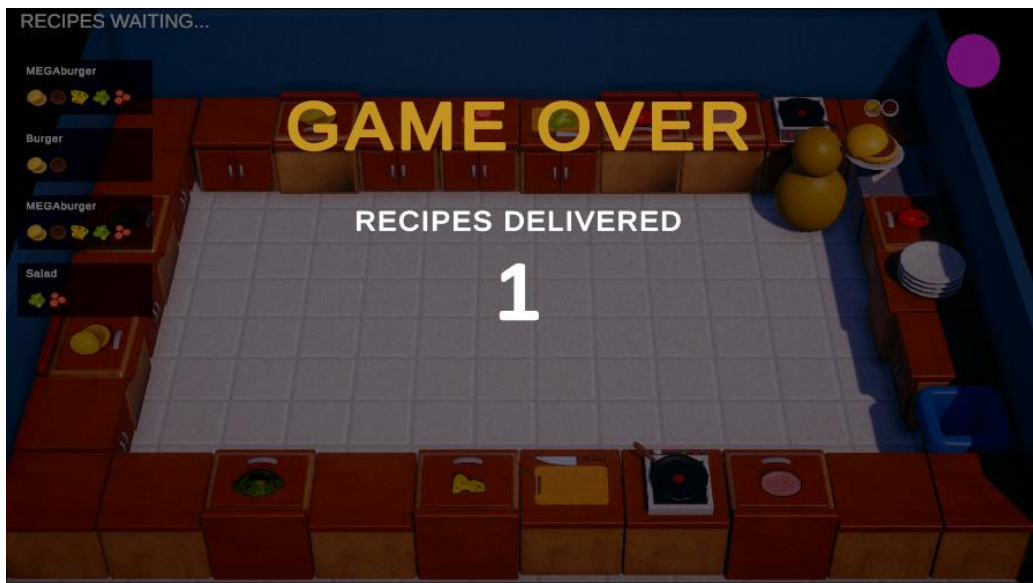


Рисунок 3.24 – Вікно “GameOver” із кількістю правильних замовлень

Таким чином, розроблено основну механіку гри у вигляді готування та видачі замовлень, дотичні до жанру cooking-симуляторів.

### 3.3.2 Методи покращення продукту, шляхи подальшого розвитку

Проект “Existence for a while” має великий потенціал у сторону покращення якості продукту, розширення механік, розробки додаткових рівнів.

Також можна глянути в бік додаткового контенту у вигляді нових тумб, зміну їхніх текстур, додавання нового виду меблів та засобів для приготування замовлень.

Ніхто не відміняв вшивання в проєкт системи мультиплеєру, що у свою чергу підтягнула б мікротранзакції в гру у вигляді продажу шкінів, як на персонажів, так і на об'єкти середовища.

Ще ніколи інді-розробники не мали таку кількість можливостей у розробці ігор, як в часі відкритості інформації, великих ком'юніті та використання стількох інструментів. Тому Unity завжди радий відкрити потенціал своїх користувчів.

### **3.4 Продюсування саунд-дизайн частини. Меню налаштувань**

Розробка Sound Design частини для проєкту “Existence for a while” заключається в додаванні SFX-ефектів навколишнього середовища та музики під час гри. Для частини із звуковими ефектами нам потрібно скористатися безкоштовним паком звуків Surface Sound із бібліотеки Unity Asset Store.

Із допомогою інструменту Unity “SoundMixer” було створено реалізацію усіх звуків у проєкті.

Створений всередині сцени об'єкт “SoundManager” із допомогою скрипту розподілятиме звуки, попередньо сформовані AudioClips, за призначенням, а ті у свою чергу будуть прикріпленні до об'єктів, умов, анімацій та процесів, що відбуватимуться у процесі гри. Таким чином добавлено: звук початкового відліку, невдачі замовлення, успіху замовлення, звуки ходьби персонажа, звук підйому та дропу предмета, звуки сковорідки, звуки смітника, та попереджувального сигналу про надмірне готування на сковороді.

Для створення головної музики відеогри, потрібно розробити її у програмі FL Studio 20.



Ключовою нотою при розробці головної музики для гри було використано тональність E-major. Інструменти, які використані при розробці музичного проєкту: Drums, Electro Guitar, Bass Guitar, Spooky Synth, Piano Synth. На рисунку 3.25 продемонстровано повністю з'єднану мелодію інструментів у середовищі FL Studio 20.

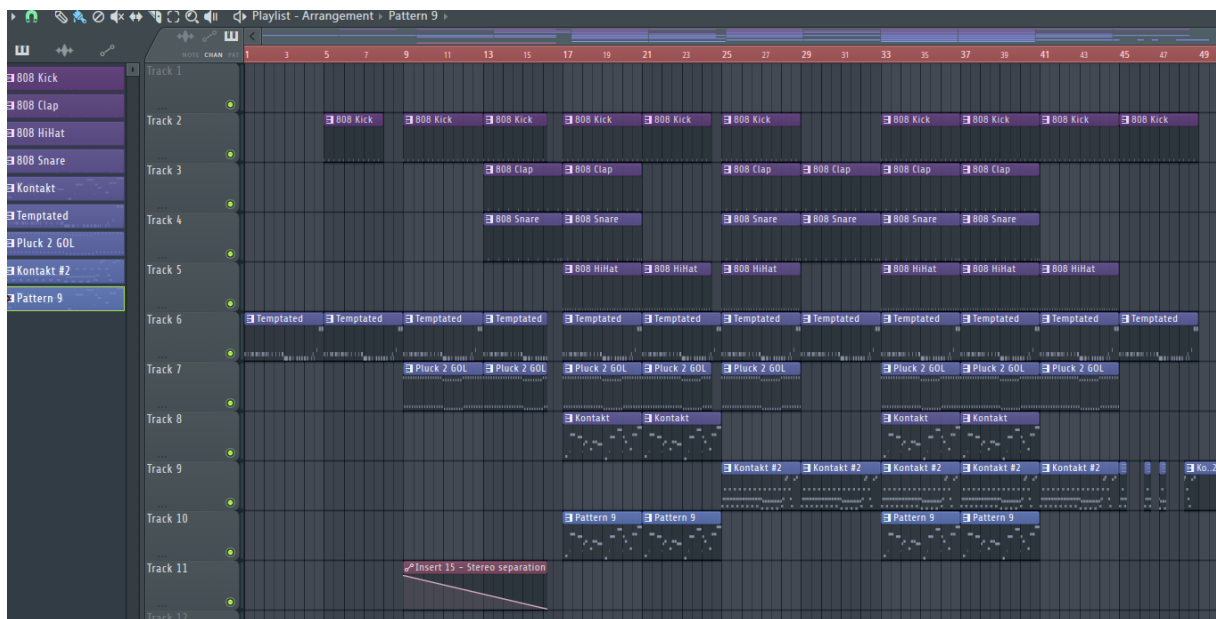


Рисунок 3.25 – Скомпонована мелодія із музичними інструментами в FL Studio

Маючи готову мелодію, експортуємо її із FL Studio у форматі MP3, імпортуємо у проєкт. Створюємо об'єкт “MusicManager” у зоні ігрової сцени і за допомогою скрипту виводимо музику в гру.

Для створення розширеного UI-меню опцій, потрібно створити кнопки зміни гучності звукових ефектів та музики, а також додати функцію зміни клавіш керування. За допомогою інструментів Unity UI створюємо потрібні елементи, робимо референси, а тоді, використовуючи скрипт “OptionsUI”, програмуємо дане вікно, що виконуватиме функцію налаштувань. На рисунку 3.26 продемонстровано вікно налаштувань відеогри “Existence for a while”, що повністю функціонує.

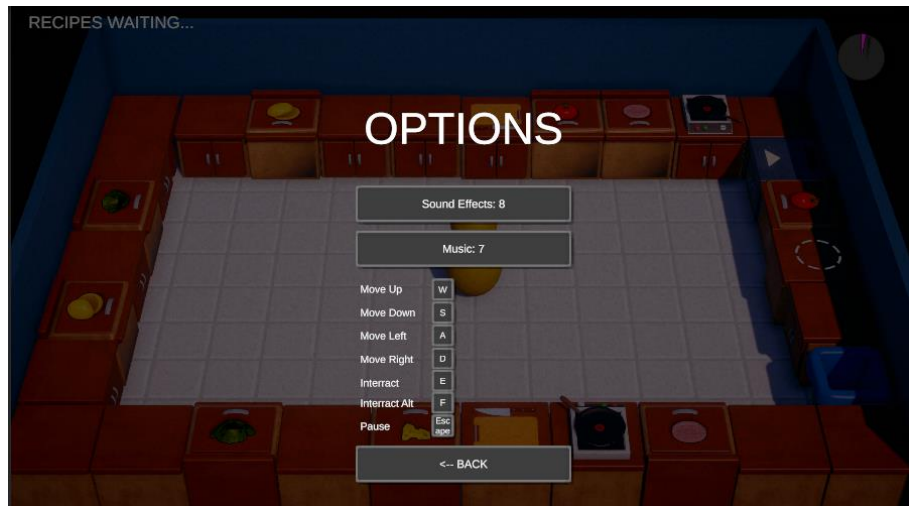


Рисунок 3.26 – Меню налаштувань у відеогрі

Коли всі компоненти готові, можна розглянути загальну архітектуру папок та об'єктів проєкту, що являється 3D відеогрою “Existence for a while”, розробленою за допомогою рушія Unity, що подана у додатку Б. Залишилось зробити білд гри і проєкт виконано.

### 3.5 Висновок до третього розділу

У третьому розділі кваліфікаційної роботи проведено розробку візуальної складової гри, на яку в першу чергу звертатиме увагу гравець при відкриванні відеогри. Створено 2D-об'єкти спрайтів, що використовувалися, як в UI-компонентах, так і слугували текстурами для об'єктів. Використовуючи середовище Blender, було створено 3D-моделі об'єктів. Розроблено анімацію головного персонажа та інших елементів середовища. Проведена робота над розробкою шейдера для “DeliveryCounter”. Розглянуто використання інструменту Cinemachine, як основної камери та проведено світлокорекцію та пост-обробку проєкту.

Розроблено ряд скриптів мовою C# та методами ООП, що відповідають за поведінку персонажа, та об'єктів навколо нього. Було створено UI-компоненти із їхніми головними інформаційно-вказівними функціями для гравця. Розроблено сцени головного меню та вікна завантаження перед початком гри.

Розглянуто головний механізм гри, що заключається у формуванні замовлень, розроблено його логіку та результати виконання чи невиконання умов замовлень. Проведено тестування розробленої механіки. Звернено увагу на методи покращення продукту та на перспективу розвитку проєкту в комерційному плані.

Наповнено проєкт звуковими ефектами із бібліотеки Unity Asset Store та розроблений власний саунд-трек для гри у середовищі FL Studio. Імплементовано розширене меню налаштувань із можливістю регулювання звуку та заміною клавіш керування при потребі.

## РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Основні положення, стан електробезпеки в Україні та дія електричного струму на людину

Згідно з чинними нормативно-правовими актами, електробезпека полягає в системі організаційних та технічних заходів та засобів, спрямованих на захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електричного поля та статичної електрики. Травми, які виникають в результаті дії електричного струму або електричної дуги на організм людини, відомі як електротравми, при чому вони можуть виникати при проходженні або без проходження струму через тіло людини, наприклад, через опіки або засліплення електричною дугою.

Термін "електротравматизм" використовується для опису сукупності електротравм [33]. За статистичними даними, українські промислові травми, пов'язані з електричним струмом, становлять близько 1% від загальної кількості виробничих травм, при цьому до 20% є смертельними. Варто відзначити, що до 80% смертей від електротравм стаються внаслідок нещасних випадків з напругою до 1000 В, оскільки низьковольтні системи електропостачання доступні практично всім працівникам. У виробничих умовах кількість електротравм становить близько 500 випадків щорічно, з них приблизно 150 призводять до смерті. Оскільки електроенергія широко використовується у всіх сферах господарства, проблема електробезпеки набуває особливого значення. Порівняно з іншими видами травм, електротравми мають кілька особливостей, так як людина не може визначити наявності напруги безпосередньо, тому дія струму зазвичай є раптовою. Крім цього, струм впливає на всі органи та системи організму, що може призвести до серйозних порушень їх функціонування.

Вплив електричного струму на організм має різноманітні прояви, включаючи термічні, електролітичні та біологічні наслідки. Термічний ефект

виявляється у нагріванні тканин, випаровуванні вологи та може призводити до опіків та розриву тканин. Електролітичний ефект полягає в розкладанні органічних рідин та порушенні їх складу. Біологічна реакція полягає в збудженні та порушенні внутрішніх процесів організму, що може виявлятися у випадковому скороченні м'язів та порушенні діяльності важливих органів, таких як серце та легені.

Електричні травми можна умовно розділити на місцеві, загальні та комбіновані [33]. До місцевих належать опіки, механічні ушкодження та електрофтальмія. Опіки є найбільш поширеними видами електротравм, які, залежно від умов, бувають контактними, дуговими або комбінованими. Контактні опіки, як правило, виникають в установках з невеликою напругою через тепловий вплив струму і пошкоджують прилеглі тканини. Дугові опіки можуть виникати внаслідок виникнення електричної дуги внаслідок короткого замикання між елементами установки або між елементами та тілом людини, що може призвести до серйозних ушкоджень і навіть смерті потерпілого. Механічні ушкодження виникають внаслідок неконтрольованих судорожних скорочень м'язів, які виникають під впливом струму. Електрофтальмія – це запалення зовнішніх оболонок очей, викликане ультрафіолетовим випромінюванням електричної дуги.

Загальні електричні травми включають в себе електричний удар, який може привести до судорог, зупинки дихання і серцевої діяльності через порушення роботи різних груп м'язів, що може призвести до втрати свідомості та навіть клінічної смерті. Комбіновані об'єднують місцеві та загальні.

#### **4.2 Ергономічні вимоги до організації робочих місць користувачів комп'ютерів**

Організація робочого простору для користувачів комп'ютерів має відповідати стандартам, визначеним у НПАОП 0.00-7.15-18 [34] та ДСТУ 8604:2015 [35]. Наприклад, площа одного робочого місця з використанням

відеодисплейного терміналу (ВДТ) повинна бути не менше 6,0 м<sup>2</sup>, а об'єм приміщення - не менше 20 м<sup>3</sup>. Робочі місця з ВДТ слід розміщувати на відстані не менше 1 м від стіни з вікнами; відстань між бічними поверхнями ВДТ має бути не менше 1,2 м; відстань між задньою поверхнею одного ВДТ та екраном іншого - не менше 2,5 м; прохід між рядами робочих місць має бути не менше одного метра. Також важливо враховувати розміри меблів (див. рисунок 4.1, взятий з [34]) для комп'ютеризованих робочих місць, таких як стіл для ВДТ, який має ширину 1200 мм і глибину 800 мм [34]. Щоб уникнути відбиття природного світла на екрані ВДТ, рекомендується розміщувати їх вздовж стіни з вікнами. Для зменшення впливу шуму з сусідніх робочих місць і забезпечення кращої концентрації під час виконання складних завдань, рекомендується використовувати перегородки висотою 1,5 - 2 м для відокремлення робочих зон.

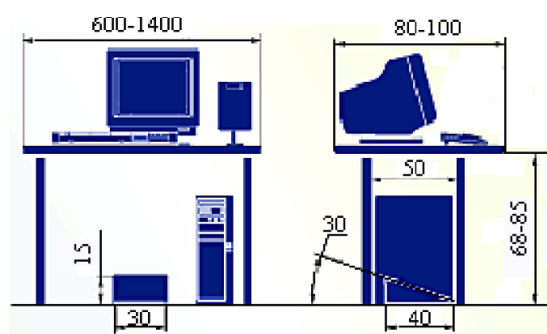


Рис. 4.1 – Організація робочого місця

Ефективне розташування відео терміналу, клавіатури та принтера на робочому місці є ключовим чинником для забезпечення безпеки та зручності користувачів комп'ютерів. Правильне розташування екрана (дисплея) повинно забезпечувати комфортне спостереження зором у вертикальній площині під кутом  $\pm 30^\circ$  від лінії зору оператора. Найкращі умови для зорового сприйняття досягаються, коли верхній край екрана знаходиться на рівні очей, а погляд спрямований вниз на центр екрана. Оскільки робота за комп'ютером найбільш комфортна з нахилом голови вперед на близько  $20^\circ$  від вертикалі (що сприяє розслабленню м'язів шиї), то екран також повинен бути нахилений назад на  $20^\circ$

від вертикалі. Рекомендована оптимальна відстань від очей користувача до екрана та клавіатури повинна бути не менше 600 мм, з урахуванням розміру символів. Наприклад, при діагоналі екрана 35 см, відстань до очей має бути 60–70 см, при діагоналі 43 см – 70 см, при діагоналі 48 см – 80 см. Для клавіатури комп'ютера на робочому столі має бути виділений достатній простір для переміщення та поворотів. Положення клавіатури і кут її нахилу повинні відповідати індивідуальним побажанням користувача. Кут нахилу клавіатури може змінюватись від 5 до 10 градусів. Якщо клавіатура не має вбудованого простору для долонь, їх слід розташовувати на відстані не менше 100 мм від краю столу в зоні оптимального моторного поля. Також можливе розташування клавіатури на спеціальній робочій поверхні, окремо від основного столу. Розташування принтера або іншого пристрою введення-виведення інформації на робочому місці повинно забезпечити зручний доступ до екрана комп'ютера і можливість комфортного керування пристроєм в межах моторного поля (висота від 900 до 1300 мм, глибина від 400 до 500 мм).

### **4.3 Організація і функціонування системи управління охороною праці**

Україна має складну систему управління охороною праці [36], яка включає різні рівні державної влади, управлінські структури підприємств і організацій, а також трудові колективи. Власне СУОП [37] поділяється на дві основні групи, а саме ланки, що забезпечують вирішення загальних питань охорони праці, включаючи законодавчо-нормативні, науково-технічні, соціально-економічні аспекти тощо, та ланки, які забезпечують безпеку праці на рівні конкретних організацій і підприємств. До перших відноситься Кабінет Міністрів України, який відповідає за впровадження державної політики у сфері охорони праці, орієнтуючись на стан безпеки праці в країні, організує розробку загальнодержавних програм для покращення цього стану, затверджує їх і контролює виконання, а також визначає функції органів виконавчої влади з

управління питаннями охорони праці та нагляду за нею. Крім даної установи ще до першої групи входить Верховна Рада України, яка співпрацює з відповідними органами державної виконавчої влади у визначенні державної політики у галузі охорони праці. Вона також розглядає питання про удосконалення та розвиток законодавства щодо охорони праці, а також соціальні аспекти, пов'язані з умовами та безпекою праці. На останок ще виділяють Держгірпромнагляд України, Фонд соціального страхування від нещасних випадків та центральні з місцевими органами влади. Перша організація втілює державну стратегію в сфері гірничого нагляду та промисловості, і співпрацює з міжнародними організаціями щодо охорони праці. Рішення установи обов'язкові для виконання всіма міністерствами, центральними органами державної влади, місцевими державними адміністраціями, радами народних депутатів і підприємствами. Фонд соціального страхування від нещасних випадків координує страхову діяльність у сфері охорони праці. Серед центральних органів влади, Міністерство праці і соціальної політики здійснює державну експертизу умов праці та контроль за якістю атестації робочих місць, коли як інші міністерства визначають політику конкретної галузі, розробляють заходи щодо безпеки праці та навчають працівників правилам охорони праці. Для забезпечення виконання вказаних завдань в апаратах міністерств і інших центральних органів державної виконавчої влади формуються служби охорони праці. Останніми з першої групи розглянемо місцеві державні адміністрації та органи місцевого самоврядування, які розробляють місцеві програми для покращення безпеки та гігієни праці, здійснюють контроль за дотриманням нормативних актів на місцевому рівні.

До другої групи належать управлінські органи підприємств, які забезпечують виконання вимог законів та нормативних актів для створення безпечних умов праці та запобігання травм і захворювань серед працівників, вирішуючи всі питання, пов'язані з виробництвом. У своїй діяльності вони



взаємодіють з комісією з охорони праці підприємства, якщо така є, профспілками та представниками трудового колективу.

Правова основа системи управління охороною праці включає в себе Конституцію України, Кодекс законів про працю України, а також законодавство про охорону праці та державне соціальне страхування від нещасних випадків на виробництві та професійних захворювань, які призвели до втрати працездатності. Крім того, до складу правової бази входять накази та розпорядження Президента України, а також акти управління, такі як розпорядження та постанови Кабінету Міністрів, Держгірпромнагляду, Міністерства охорони здоров'я, Міністерства праці і соціальної політики, а також інших органів, які мають директивні повноваження з питань охорони праці.

Вихідні параметри СУОП визначаються на основі вимог нормативних документів, аналізу виробничої ситуації та інших факторів, тому їх можна віднести до категорії багатоконтурних систем, які піддаються програмуванню. Це пояснюється складністю об'єкта управління та його великою інерційністю.

При роботі в Unity під час розробки відеогри, яка розглянута у результаті дослідження, користувачі використовуватимуть комп'ютер, тому необхідно обговорити питання електробезпеки при роботі з цим обладнанням. Головна мета електробезпеки полягає в зменшенні ризику негативного впливу електричного струму на людину. Цій меті можна досягти шляхом:

- створення безпечних та надійних конструкцій електроустановок;
- застосування організаційних та технічних заходів для безпечної експлуатації електроустановок та використання електричної енергії;
- використання технічних засобів захисту.

Залежно від засобів електробезпеки, усі електротехнічні вироби поділяються на 5 класів:

- Клас 0 – електрична установка має лише робочу ізоляцію як засіб захисту.
- Клас I – має як ізоляцію, так і заземлення. Клас 0I проміжний.

- Клас II забезпечує подвійну ізоляцію.
- Клас III передбачає використання лише низької напруги (до 42 В) для живлення установки.

Комп'ютер швидше представляє собою конструкцію з кількох таких пристроїв, а тому його не можна віднести до однієї групи в цілому. Але якщо розглянути конкретно саму найнебезпечнішу частину, а саме блок живлення, то його можна віднести до класу II. Стан ізоляції провідних частин системи забору повинен відповідати вимогам "Правил використання електроустановок". Згідно з цими правилами, передбачено регулярне перевірка ізоляції: двічі на рік у приміщеннях зі складними умовами і підвищеною вологістю, та один раз на рік у приміщеннях з нормальним середовищем. Ізоляція створює великий опір, що перешкоджає протіканню струму через неї. Опір ізоляції кожної системи повинен бути не менше 0,5 МОм (500 000 Ом). У разі зниження опору ізоляції на 50% від початкового, ізоляцію потрібно замінити.

#### **4.4 Висновок до четвертого розділу**

В цілому підсумовуючи про системи управління охороною праці, то на конкретній організації вона завжди є багаторівневою, де верхній рівень представлений державним управлінням, а нижній – управлінням на конкретному об'єкті. Крім цього, описано ергономічні вимоги щодо комп'ютерного робочого місця та як правильно його облаштувати в залежності від пристроїв. Так наприклад чим більшою йде діагональ монітора, тим на більшу відстань необхідно сидіти для збереження здоров'я очей. Також, в цьому розділі піднято питання та подано інформацію щодо електробезпеки при роботі з комп'ютерним устаткуванням, та про можливі електротравми, які можуть виникнути.

## ВИСНОВКИ

Внаслідок проходження циклу розробки комп'ютерної гри "Existence for a while" у 3D-середовищі із використанням ігрового рушія Unity, було отримано хороший фінальний продукт у жанрі cooking-симулятора. Було розроблено кожен компонент гри за допомогою стороннього спеціалізованого програмного забезпечення, а також із використанням бібліотеки Unity Asset Store, а згодом із допомогою рушія Unity скомпоновано у готовий до виконання проєкт. Система рівнів, та мультиплеєрна складова могла б значно краще привернути увагу до гри, тож проєкт залишається необмеженим, але і в такому вигляді є інструментом для розваг.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Подано аналіз ринку відеоігор та їх популяризація із кожним роком.
- Розглянуто список програмного забезпечення необхідного для розробки.
- Висвітлено Основний спектр інструментів при роботі із 3D рушієм Unity.
- Проаналізовано підхід для взаємодії інструментів та їх імплементації у майбутній проєкт.

В другому розділі кваліфікаційної роботи:

- Досліджено успіх відеоігор у жанрі cooking симуляторів та в чому їхня головна ідея і конкурентоспроможність на ринку проєктів під іншими жанрами.
- Обґрунтовано причини використання допоміжної бібліотеки Unity Asset Store як засіб для пришвидшення процесу розробки.
- Сформовано чіткі вимоги у проєктуванні та створено прототипи майбутніх об'єктів майбутньої відеогри на базі 3D рушія Unity.

В третьому розділі кваліфікаційної роботи:

- Розроблено кожен важливий компонент відеогри, що в сукупності дає інформаційний продукт, готовий до тестування та шліфування.

- Запропоновано методи покращення відеогри в майбутньому аби привернути більше уваги ком'юніті і, можливих майбутніх інвесторів.
- Протестовано всі розроблені логічні взаємодії та механіки між гравцем та об'єктами та функціонал меню налаштувань.
- Довершено візуальний вигляд за допомогою пост-обробки та звуковий супровід із допомогою саунд-дизайну.

В останньому розділі проаналізовано поняття електробезпеки та різні аспекти дії електричного струму на живу тканину, включаючи термічну, електролітичну та біологічну дію, а також види електротравм і їх наслідки. Також описано функціонування СУОП та підкреслено важливість правильного розташування компонентів робочого місця, таких як екран, клавіатура та принтер, для забезпечення комфорту та запобігання захворювань.

## ПЕРЕЛІК ДЖЕРЕЛ

- 1 IT Step (2021). Мови програмування, на яких написані популярні комп'ютерні ігри. 22 січня. Доступно: [https://kam.itstep.org/blog\\_3/programming-languages-in-which-popular-computer-games-are-written](https://kam.itstep.org/blog_3/programming-languages-in-which-popular-computer-games-are-written) [Дата звернення: 2 травня 2024].
- 2 The statistics portal (2024). Annual gaming company investment volume. Retrieved by J. Clement May 8, 2024. Available at: <https://www.statista.com/statistics/1452456/annual-gaming-company-investment-volume/> [Дата звернення: 2 травня 2024].
- 3 Проект Еразмус+ (2019). Основи комп'ютерних ігор та ігрових програм : довідник модуля. / С. О. Нікітін, Л. О. Нікітіна. – Х. : «Друкарня Мадрид», 2019. – 138 с. Доступно: [https://web.kpi.kharkov.ua/gamehub/wp-content/uploads/sites/230/2021/11/BCG\\_ukr.pdf](https://web.kpi.kharkov.ua/gamehub/wp-content/uploads/sites/230/2021/11/BCG_ukr.pdf) [Дата звернення: 2 травня 2024].
- 4 Strutynska, H. Kozbur, L. Dmytrotsa, O. Hlado, I. Kozbur, N. Gashchyn: Analysis of the SMEs' Digitalization State Using HIT Index and Machine Learning Technique. 13th International Conference on Advanced Computer Information Technologies (ACIT). Publisher: IEEE. Institute of Electrical and Electronics Engineers Inc. Wroclaw, Poland. - p. 332-337 (Scopus). Available at: <https://ieeexplore.ieee.org/document/10275519> [Дата звернення: 2 травня 2024].
- 5 London Product Academy (2023). Моделі монетизації продукту: від безкоштовних до платних моделей. 13 серпня. Доступно: <https://www.londonproduct.academy/post/modeli-monetizaciyi-produktu-vid-bezkoshtovnih-do-platnih-modeley> [Дата звернення: 3 травня 2024].
- 6 I. Strutynska, L. Dmytrotsa, H. Kozbur, L. Melnyk, R. Sherstiuk: The Unification of Approaches to Measuring the Digital Maturity of Business Structures (International and Domestic Approaches Volume I: Main Conference, PhD Symposium, and Posters, Kherson, Ukraine, September 28 - October 2, 2021. CEUR Workshop Proceedings. ICTERI 2021: pp. 10-23. Available at: <https://ceur-ws.org/Vol-3013/20210010.pdf> [Дата звернення: 3 травня 2024].

7 I. Strutynska, L. Dmytrotsa, H. Kozbur, L. Melnyk: The Digital Business Transformation Index Determining and Monitoring: Development of a National Online Platform. Theoretical and Applied Problems, Ternopil, Ukraine, November 16-18, 2021. CEUR Workshop Proceedings 3039, CEUR-WS.org ITТАР 2021: pp. 327-334. Available at: <https://ceur-ws.org/Vol-3039/short33.pdf> [Дата звернення: 5 травня 2024].

8 Newzoo (2020). The World's 2.7 Billion Gamers Will Spend \$159.3 Billion on Games in 2020. 8 травня. Доступно: <https://newzoo.com/resources/blog/newzoo-games-market-numbers-revenues-and-audience-2020-2023> [Дата звернення: 5 травня 2024].

9 Бібічков, І. (2023). Як розробляють ігри? 4 вересня. Доступно: <https://lemon.school/blog/yak-rozroblyayut-igry> [Дата звернення: 5 травня 2024].

10 Autodesk (2024). Autodesk 3ds Max. Доступно: <https://www.autodesk.com/products/3ds-max/overview?term=1-YEAR&tab=subscription> [Дата звернення: 5 травня 2024].

11 Blender Foundation. (2024). About Blender. Доступно: <https://www.blender.org/about/> [Дата звернення: 5 травня 2024].

12 Business.com. (2024). Photoshop Reigns Supreme: How the Software Has Maintained Market Dominance. Доступно: <https://www.business.com/articles/photoshop-reigns-supreme-how-the-software-has-maintained-market-dominance/> [Дата звернення: 6 травня 2024].

13 Noble Desktop. (2024). What is Figma? Доступно: <https://www.nobledesktop.com/learn/figma/what-is-figma> [Дата звернення: 6 травня 2024].

14 PCMag. (2024). Image Line FL Studio Review by Jamie Lendino. Доступно: <https://www.pcmag.com/reviews/image-line-fl-studio> [Дата звернення: 6 травня 2024].

15 Unity Technologies. (2024). Unity. Доступно: <https://unity.com/> [Дата звернення: 8 травня 2024].

16 SumDU uLab. (2022). 10 найкращих ігрових рушіїв. Доступно: <https://ulab.sumdu.edu.ua/uk/10-najkrashhih-igrovih-rushiiv> [Дата звернення: 8 травня 2024].

17 Unity Technologies. (2024). Tools. Доступно: <https://assetstore.unity.com/tools> [Дата звернення: 9 травня 2024].

18 Unity Technologies. (2024). Unity User Manual. Доступно: <https://docs.unity3d.com/Manual/> [Дата звернення: 9 травня 2024].

19 Skvot.io. (2022). Історія геймдеву ХХ століття: від Spacewar! до Sonic the Hedgehog. Доступно: <https://skvot.io/uk/blog/istoriya-geymdevu-hh-stolittya-vid-spacewar-do-sonic-the-hedgehog> [Дата звернення: 9 травня 2024].

20 Sandbox. (2024). Cooking Simulator. Доступно: <https://docs.sandbox.game/en/v/creators/game-maker/creative-guides/experience-guide/resource-management/cooking-simulator> [Дата звернення: 12 травня 2024].

21 Unity Technologies. (2020). What is the Unity Asset Store and How to Make Money on It. Доступно: <https://forum.unity.com/threads/what-is-the-unity-asset-store-and-how-to-make-money-on-it.999193/> [Дата звернення: 12 травня 2024].

22 Лугова Т.А., Блажко О.А. Проектування комп'ютерних ігор для навчання : навчальний підручник / Т.А. Лугова, О.А. Блажко. – Одеса : ФОП «Побута». – 2019. – 122 с.

23 UnivDatos. Патні, Х. (2022). Simulation Games Market. Доступно: <https://univdatos.com/uk/simulation-games-market/> [Дата звернення: 15 травня 2024].

24 Occasoftware. Cubicle, L. (2023). Unity: What is a Scene? Доступно: <https://www.occasoftware.com/blog/unity-what-is-a-scene> [Дата звернення: 15 травня 2024].

25 Kas, S. (2020). How Does Camera Perspective Affect Player Experience and Feelings? Доступно: <https://medium.com/@staskas/how-does-camera-perspective-affect-player-experience-and-feelings-b187e6824f7c> [Дата звернення: 16 травня 2024].

26 Retro Style Games. (2023). Game User Interface Design Examples. Доступно: <https://retrostylegames.com/blog/game-user-interface-design-examples/> [Дата звернення: 16 травня 2024].

27 Pixune. Mehrafrooz, B. (2024). Game HUD Design. Доступно: <https://pixune.com/blog/game-hud-design/> [Дата звернення: 16 травня 2024].

28 Giant Bomb. (2023). Options Menu. Доступно: <https://www.giantbomb.com/options-menu/3015-6644/> [Дата звернення: 16 травня 2024].

29 80.lv. Sergeev, A. (2020). Defining Environment Language for Video Games. Доступно: <https://80.lv/articles/defining-environment-language-for-video-games/> [Дата звернення: 16 травня 2024].

30 FGFactory. (2023). Exploring Graphics: Visual Styles in Games. Доступно: <https://fgfactory.com/exploring-graphics-visual-styles-in-games>

31 Game Developer. Naisan, M. (2021). The Logo Design Process - The Steps Involved in Designing a Logo for Your Game. Доступно: <https://www.gamedeveloper.com/business/the-logo-design-process---the-steps-involved-in-designing-a-logo-for-your-game> [Дата звернення: 18 травня 2024].

32 Kevuru Games. (2023, грудень 10). What is Animation in Video Games: Basic Principles and Applications. Alena Porokh. Доступно: <https://kevurugames.com/blog/what-is-animation-in-video-games-basic-principles/> [Дата звернення: 18 травня 2024].

33 Безпека життєдіяльності [текст]: підручник. / [О. І. Запорожець, Б. Д. Халмурадов, В. І. Применко та ін.] – К.: «Центр учбової літератури», 2013. – 448 с. [Дата звернення: 18 травня 2024].

34 Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями : Наказ Міністерства соц. політики України від 14.02.2018 р. № 207. Доступно: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text> [Дата звернення: 26 травня 2024].



35 ДСТУ 8604:2015: Наказ Міністерства соц. політики України від 14.02.2017 р. № 204. Доступно: <https://zakon.rada.gov.ua/rada/show/v0204774-15#Text> [Дата звернення: 26 травня 2024].

36 Охорона праці в галузі інформаційних технологій : навч. посіб. / В.І. Голінько, М.Ю. Іконніков, Я.Я. Лебедєв ; М-во освіти і науки України, Нац. гірн. ун-т. – Д. : НГУ, 2015. – 246 с. [Дата звернення: 26 травня 2024].

37 Охорона праці [текст] : навчальний посібник / П. С. Атаманчук, В. В. Мендерецький, О. П. Панчук, Р. М. Білий - К. : «Центр учбової літератури», 2017. - 322 с. [Дата звернення: 26 травня 2024].

# ДОДАТКИ

## Фінальний вигляд міні-тutorіалу для гри “Existence for a while”

